Phase Projection Using Three Satellites

by

Michael C. Yeung

B.S., Electrical Engineering and Computer Science (2005)

University of California, Berkeley

Submitted to the Department of Electrical Engineering and Computer Science
In Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE
IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2007

Signature of Author.................................. ....
Department of Elec mputer Science
May 11, 2007

Certified by................ ....
Jin Au Kong
F of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by......................... ....
Bae-Ian Wu
rch Scientist
-Supervisor

Accepted by.............. ....
Arthur C. Smith
Chairman, Committee on Graduate Students
Department of Electrical Engineering and Computer Science

Phase Projection Using Three Satellites

by

Michael C. Yeung

Submitted to the Department of Electrical Engineering
on May 11, 2007 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Electrical Engineering and Computer Science

ABSTRACT

This study seeks to investigate various techniques used in Interferometric Synthetic Aperture Radar (InSAR) during the phase unwrapping process and the noise filtering step. In particular, as intuition would follow, we establish that the advantage of using three satellites over using just two in noise reduction means that using three satellites should be the way to go in all future work. In addition, this thesis does some in-depth investigation into the effectiveness of the following phase unwrapping methods: weighted least-squares phase unwrapping, branch-cut, and MAP dynamic programming phase unwrapping. The weighted least-squares unwrapping utilizes residues, which are points of inconsistencies on the interferogram, as a guide to assign weights when doing the weighting before the actual unwrapping step. Results show that weighted least square unwrapping fares the best in terms of mean RMS height error.

We also introduce the notion of 2-D and 3-D Projection in our search for a simple, elegant solution to further reduce noise during our InSAR post-processing steps. 2-D Projection is shown to be a special case of 3-D Projection, and both require the presence of at least 3 satellites. Projection in general reduces noise in the system by utilizing the information provided by the configuration of the satellites to reduce the set of allowed phase points, thereby improving the robustness of the system to noise. Our results show that, for almost all cases, whether in the extremely small baseline distance or non-integer baseline ratios, using 3-D Projection gives much better results.

Results are entirely simulation based, using the engineering tool MATLAB Version 7.0 developed by Mathworks. Root-mean-square errors will be used as the metric for comparison for most cases.

Thesis Supervisor: Jin Au Kong
Title: Professor of Electrical Engineering

Thesis Co-Supervisor: Bae-Ian Wu
Title: Research Scientist

Acknowledgements

Regardless of what the cover page might say, I did not accomplish the writing of this thesis alone, and there are many other people who unknowingly have contributed to the successful completion of this research, and to making me the person I am today.

Firstly, I would like to thank Professor Jin Au Kong for his wisdom and insight, and Bae-Ian Wu for always giving great advice on the research direction. Of course, many thanks go to Dr. Hara and Mitsubishi Corporation for allowing me to embark on this project in the first place. In addition, I have had the privilege of meeting some wonderful friends in CETA: Song Liang, Baile, Brandon, Bill, Beijia, Dongxing, Xiang Xiang, and Kei. You guys will be missed.

I dearly thank my Mom and Dad for raising me and shaping me to be the man I am. The constant phone calls, the words of worry and anxiety, and the repetitive nagging all showed how much you care. And my sisters, Wendy and Connie – though we are separated by age and distance, I know that I will always have your support in every situation, and just having that very thought is strength enough for me to draw from.

Yijia, Ivan, and Mel – you have seen me through so much. It has been so long since we last met, and time inevitably has brought us along different paths that do not look like intersecting anytime soon. I know that you will always be available to listen to any complaints I have, and you know what I am going to say before words even come out of my mouth. I thank God everyday for bringing you into my life, because I knew back then, and I still know now, that you will be available whenever I need you. The completion of this thesis would not have been possible without all the late-night encouragements.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Introduction to SAR Interferometry (InSAR)

Synthetic Aperture Radar (SAR) techniques have been used in many different but related fields since it was first developed in the 1950's. The varying range of information that SAR technology provides includes terrain structural information to geologists for mineral mining, oil spill boundaries to environmentalists, ice hazard maps to sea navigators, and reconnaissance and stealth information to military personnel, just to name a few. Interferometric SAR, or InSAR, is the branch of SAR that deals with using more than one SAR image to obtain additional information on the target. The phase maps that are generated from the phase differences between different pairs of SAR satellites are commonly known as interferograms. The phase differences are the key component in determining the actual three-dimensional height map (also known as the Digital Elevation Model, or DEM) of the terrain. As such, much study has been done on the recovery of terrain information from these interferograms.

An example of an interferogram is provided below, Figure 1.1. Its unwrapped counterpart is presented in Figure 1.2.

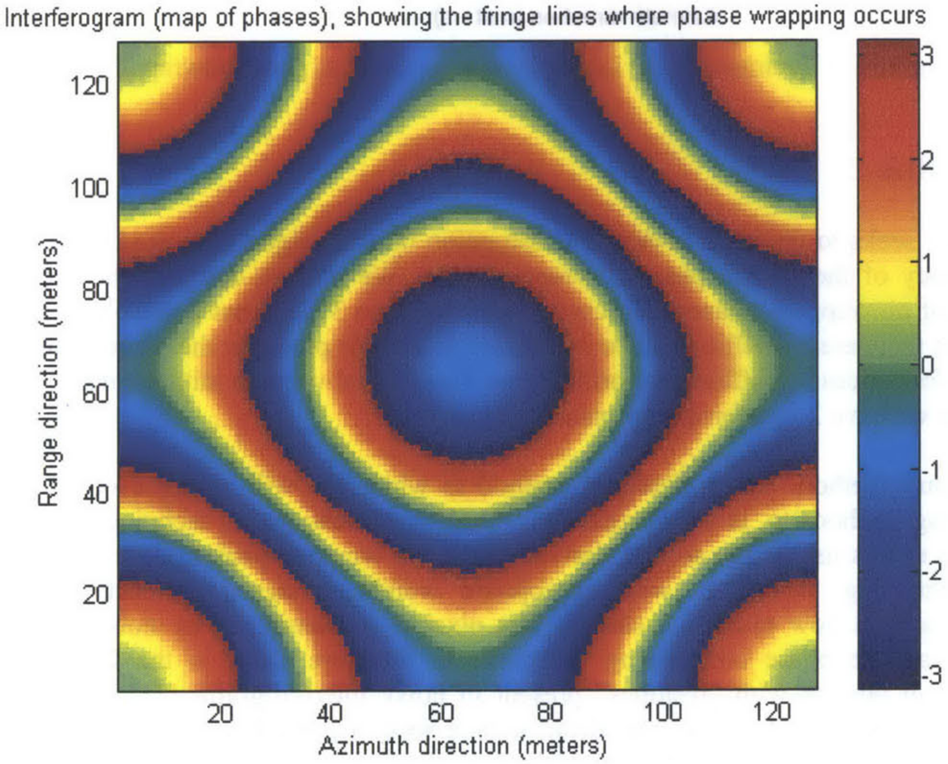

Interferogram (map of phases), showing the fringe lines where phase wrapping occurs

Figure 1.1: Example of an interferogram.

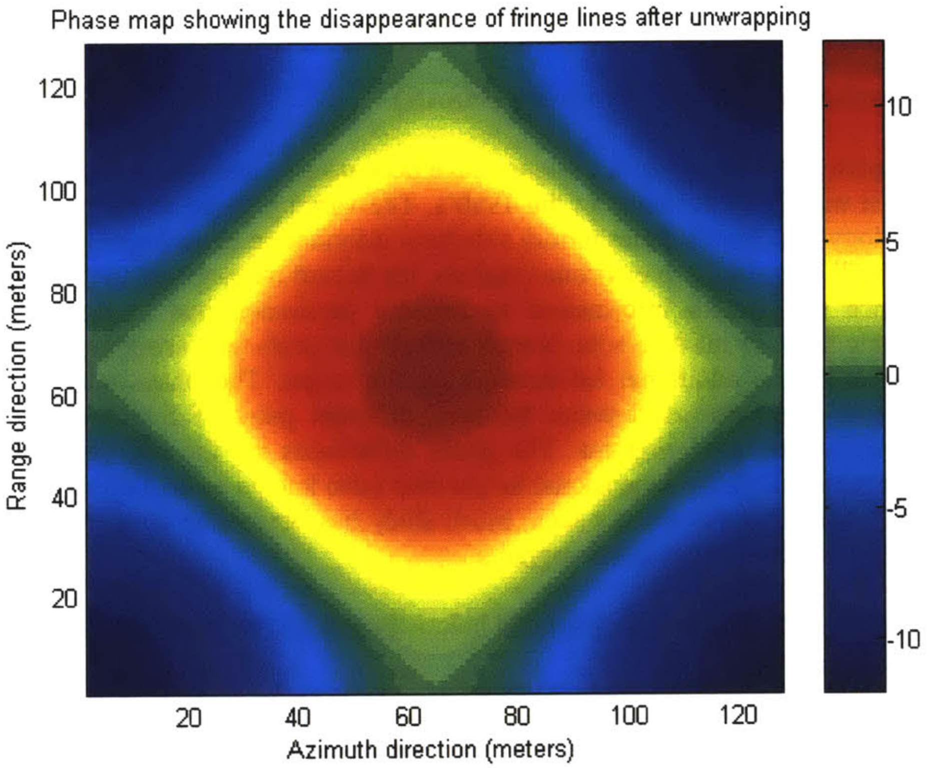


Figure 1.2: Example of the interferogram in 1.1 after phase unwrapping.

1.2 Objectives

This thesis seeks to do some in-depth investigation into the algorithms used to improve the accuracy of the critical phase unwrapping step. We seek to compare the different methods of unwrapping phases, especially when we have 3 baselines available, as is the case when there are 3 satellites. Since robustness to noise is probably the biggest factor in weak performances by different algorithms, we also look at ways to mitigate noise levels given that we have 3 interferograms instead of just 1.

The primary method, and "reference" method to which we will compare other phase unwrapping methods to, is the Weighted Least Squares unwrapping technique [1]. This technique makes use of the second derivatives of the wrapped phases to determine the best unwrapping procedure, with appropriate weights assigned to each pixel in accordance with how much we suspect that pixel is contaminated by noise. Another widely available phase unwrapping technique in literature today is the Branch Cut algorithm in tandem with "residues" present in noisy interferograms [11]. Both these methods will be presented in greater detail in later sections, and their merits compared in a test case. It should be noted that both these methods are meant for only one baseline (instead of multiple baselines), but they set up a solid framework for our multi-baseline study.

In addition, we give a thorough treatment of the iterative dynamic programming approach that Ying, Munson, Koetter, and Frey used for both the single baseline [27] and the multibaseline [21] case. This Maximum *a posteriori* (MAP) estimation that they undertook is then compared with both the single baseline and multi-baseline cases that can be achieved using Weighted Least Squares or Branch Cut, and the merits of each discussed.

We propose 3-D projection as a technique to reduce the impact of noise during our phase unwrapping step. This makes use of fact that the geometry of the satellite configuration constrains the relationships among the 3 phases retrieved by the 3 satellites. The relative distances of the satellites from one another (baseline lengths) play a critical role in this, and we lay the groundwork as to how this can be exploited to retrieve more accurate phase values.

1.3 Problem of interest

A problem inherent in retrieving information from phases and phase-related information, e.g the DEM, is the phenomenon known as phase unwrapping. Phases are naturally ambiguous in nature – they are always measured as modulo 2π. There is no observable difference between 1.5 radians and $1.5 + 8\pi$ radians – they all map to the same point on the unit circle in the complex plane, and this leads to fringe lines on interferograms. However, the terrain information depends on the actually phases, not the wrapped phases. Phase unwrapping hence becomes a critical part of the height-retrieval process. To compound the problem, noisy data may cause unnecessary phase jumps (i.e. from 0 to 2π). Since many algorithms typically use phase information from the previous pixel to unwrap the next pixel, errors induced from a phase pixel unwrapped incorrectly will quickly propagate to cause large global errors. Many phase unwrapping algorithms have been developed over the years, and most of these involve just 2 satellites. In this thesis, we seek to investigate the merits of using 3 satellites instead of 2 in the phase unwrapping process, and explore the possible algorithms that hold promise for such a critical step in height retrieval.

Moreover, the presence of 3 satellites gives us a whole new dimension to work with. Before, when there are only 2 satellites, there has never been much of a solution for checking for the validity of the data in the presence of noise. There just isn't an extra degree of freedom that can help us narrow down the effects of noise. However, when we introduce an extra satellite, the very position of this third satellite should give us information on whether the unwrapped phase data is coherent. This promises to be a powerful tool to mitigate noise in our system: since noise is assumed to be randomly distributed but data is not, the partial effects of noise might be picked up and corrected more easily if we use 3 satellites instead of 2. A big portion of this thesis will be devoted to investigating this work.

2. Fundamentals of SAR Interferometry

2.1 Configuration of satellite system

To perform any analysis and enhancements of SAR interferometry (InSAR), it is critical that we first establish the basis of the SAR system. In particular, we have to answer questions pertaining to how Synthetic Aperture Radar (SAR) is different from other conventional radar configurations, and the advantages that SAR holds over these regular methods.

Consider Figure 2.1 for a basic SAR system.

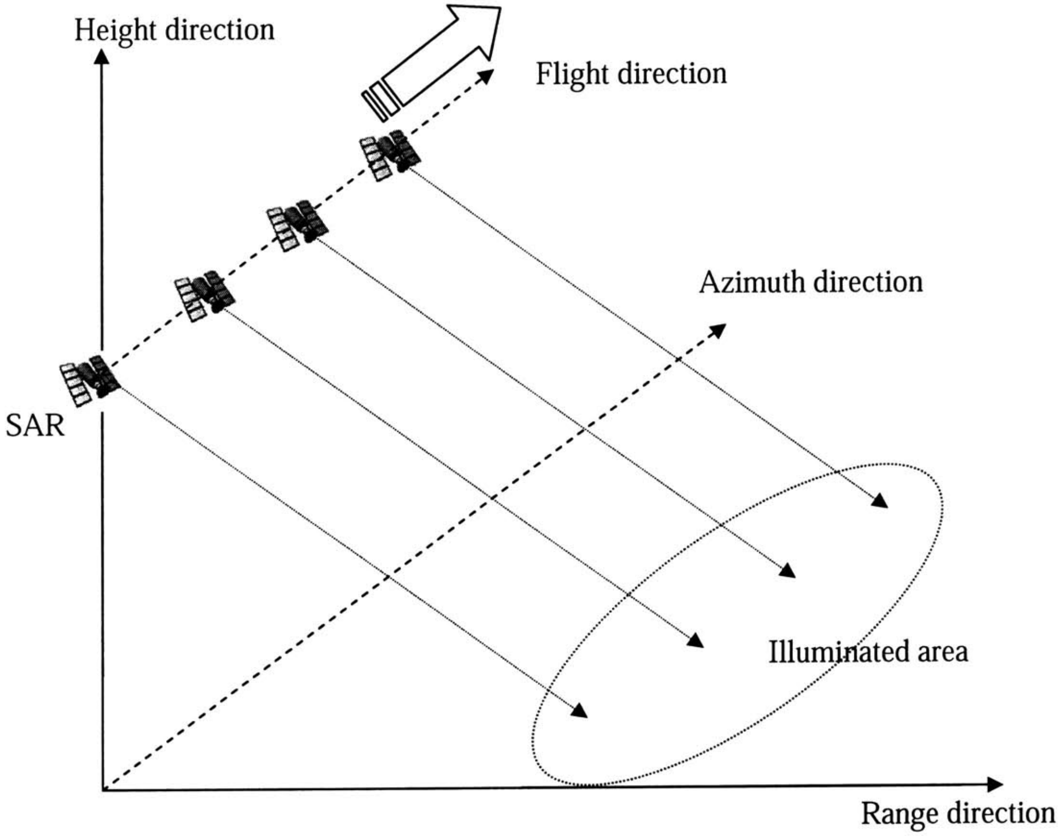


Figure 2.1: Basic operation of a SAR system.

The Synthetic Aperture Radar derives its name from its operation. A single satellite, or radar, flies over a wide azimuth distance while mapping the terrain in the range direction. This continuous flying and mapping by the satellite mimics an array of antennas, and essentially performs the same function as well. In essence, a larger, artificial (or synthetic) aperture has been replicated with the use of just one satellite in the radar system. The antenna is usually attached to either an airplane or an air-borne satellite, and this antenna serves both the purposes of transmitting and receiving signals. The frequency of the signals sent (and received) is in the microwave region [43], and can be taken to be a constant train of impulses. It is also important to note that the satellite is flying over the Earth at a much faster rate than the Earth is spinning. This non-geosynchronous orbit allows us to make the approximation that measured flight-speed of the SAR system is fully with respect to a stationary Earth terrain. Hence, for the remainder of this thesis, the effect of the time differences of arrival of signals is ignored. To simplify things a little further, the curvature of the Earth also been ignored.

Although Figure 2.1 only showed 1 satellite in motion, InSAR always involves more than just 1 satellite. This is due to the very nature which InSAR works, since there can be no interferometry when there is only one satellite present in the system. Furthermore, the equivalent signal values returning to each of the 2 SAR systems can be expressed in their real and imaginary parts, or more commonly, in their magnitude and phase. Therefore, for every pixel on the interferogram corresponding to the terrain being mapped, we can express the signal as $Ae^{j\phi}$, where A is the magnitude of the signal sent, and ϕ is the corresponding signal phase. In the case where there are 2 satellites, each satellite will receive a different signal, $A_1e^{j\phi_1}$ and $A_2e^{j\phi_2}$ for each pixel on the ground. With these 2 complex variables, let's define a new variable below:

$$A_{coh}e^{j\phi_{\text{int}}} = \left(A_1e^{j\phi_1}\right)\cdot\left(A_2e^{j\phi_2}\right)^* \tag{2.1}$$

We can thus immediately define the following terms:

$$A_{coh} = A_1 \cdot A_2 \tag{2.2}$$

$$\phi_{\text{int}} = \phi_1 - \phi_2 \tag{2.3}$$

Each of the 2 parts of the term illustrates a fundamental part of the retrieved information. They both describe the relationship between the two phase values of the same pixel on the ground due to the two satellites that make up the InSAR system. The magnitude of the term, A_{coh} describes the degree of reliability, or coherence, between the 2 pixel images. A high coherence value would indicate that the corresponding phase estimates used for height retrieval are acceptable and good height estimates are thus forthcoming. Conversely, if the A_{coh} turns out to be near zero value, the height estimate retrieved has to be considered doubtful. For the purposes of this research and thesis, we will not take into account the cases where retrieved phases are anything but perfectly coherent, i.e. A_{coh} is 1. However, it is important to note that this is one critical factor to be taken into account when applying techniques developed in this thesis into real-life data.

The second part of the term, ϕ_{int}, is defined to be the difference in phase value between 2 different satellites. As we shall see later, this is perhaps the single most important piece of information that can be retrieved from a pair of satellites, because it is directly proportional to the height of the terrain at that pixel. One important caveat of using phase to retrieve height is that phase values are inherently ambiguous. A retrieved phase value of 1.25π can really mean any value from the set of $\{(1.25 + 2n)\pi \mid n \in Integers\}$, because phases can really only exist between 0 and 2π. This means that the process of *phase unwrapping* is vital and has to be done correctly if we are to get the actual phase values, and hence the actual terrain heights, back from ϕ_{int}.

The entire process of retrieving height from SAR data is described below as a series of steps. Since there simply is no way to give each step its full deserved treatment in just one thesis, we will ignore quite a few processes in this thesis, including getting the SAR data, data co-registration, and coherence determination, working on the assumption that these ignored steps worked perfectly in a noiseless fashion. They do not lie in our scope of objectives, and since the steps listed below are fairly modular processes, i.e. one step does not interfere with the next, we can just focus on the steps that are stated in our objectives, i.e. Noise filtering, phase unwrapping, and DEM deduction.

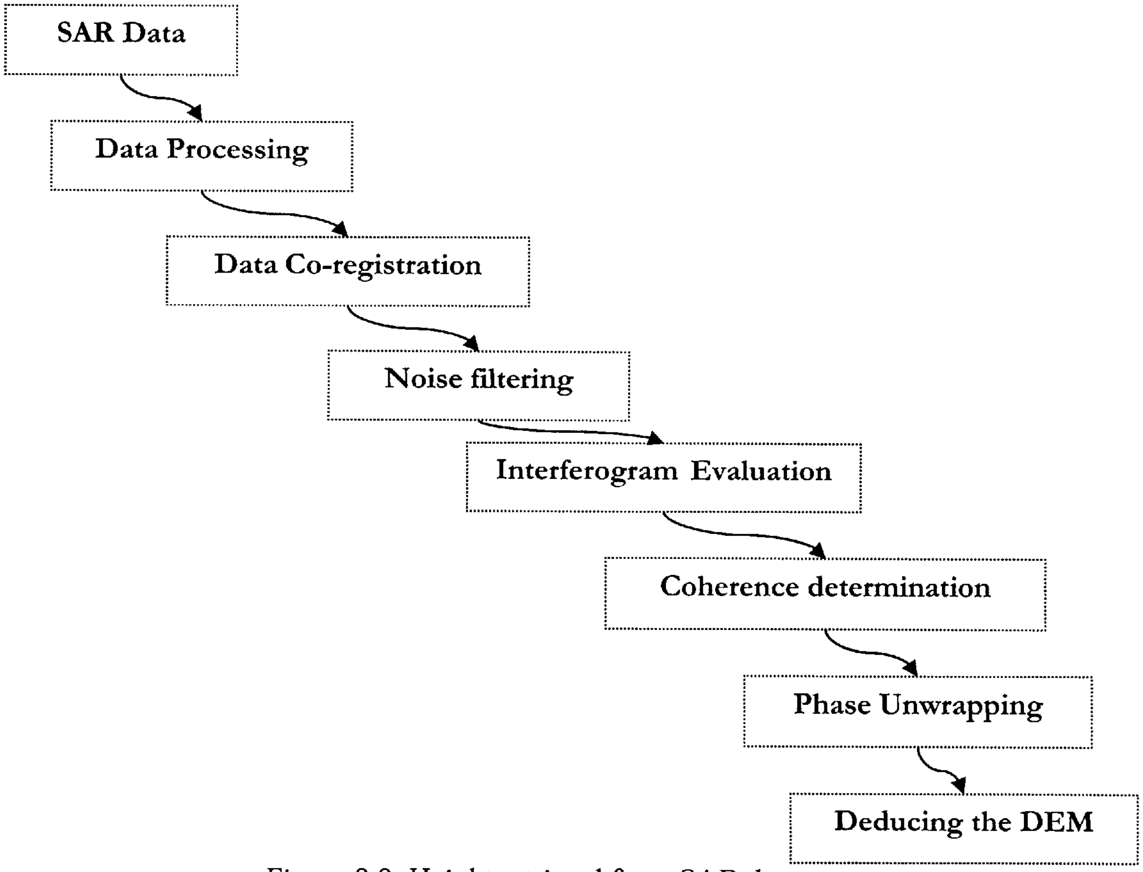


Figure 2.2: Height retrieval from SAR data.

2.2 Simulating the interferogram with a 2 satellite configuration

In this section, we will explore the exact mathematics that needs to be employed to retrieve height information with phases for the case of 2 satellites. Since the multi-satellite case of N satellites can really be thought of as being a vast system of $\frac{N!}{2\cdot(N-2)!}$ pairs of satellites, the mathematics that we are going to present can easily be extended with no loss of generality.

We start with a 1-dimensional view of the terrain, and seek to obtain the phase difference between the 2 satellites. At every azimuthal (cross-range) pixel interval, a shot of all the pixels in the range direction is taken, and stitching together all the cross range pixels generates a 2-dimensional phase map of the terrain. We assume we only know (or can measure) the following system parameters: baseline length B, baseline slant angle α, slant range of the first satellite ρ_1, wavelength of the signals transmitted received λ, and the distance of the first satellite from the ground, H. We are able to find (by transmitting and receiving the pulses) the wrapped phases of SAR 1 and 2. Figure 2.3 shows the SAR setup, as well as the corresponding system parameters.

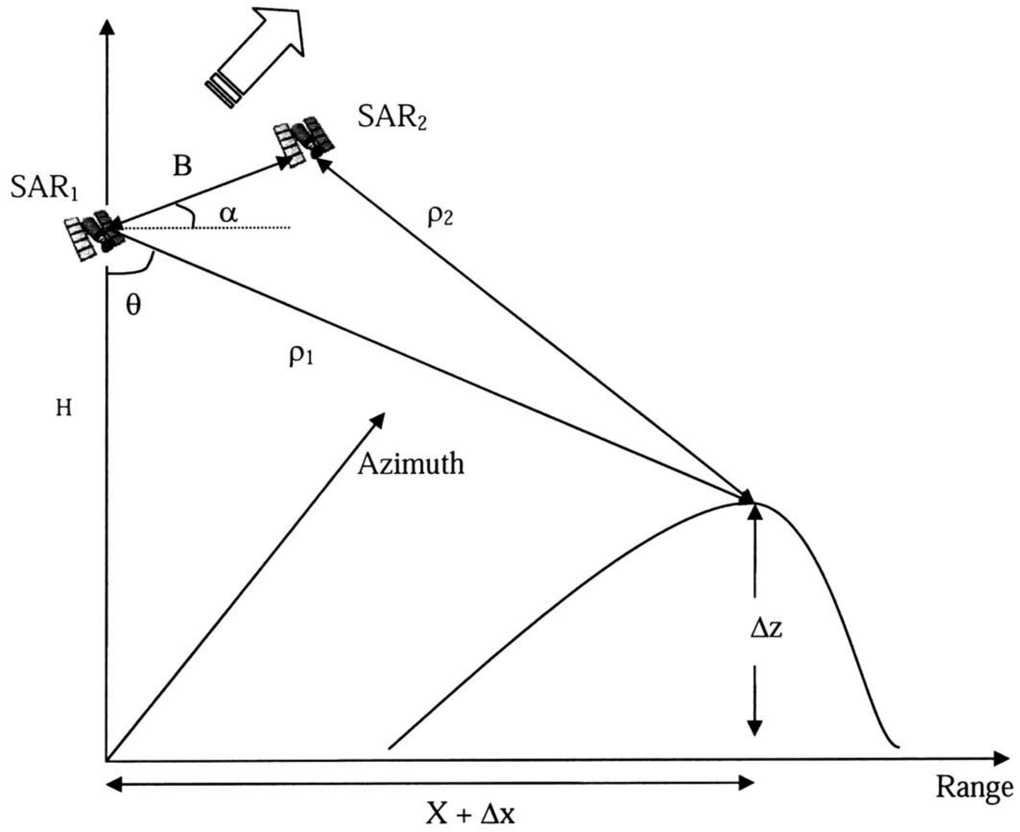


Figure 2.3: InSAR configuration for 2 satellites.

We can first obtain the slant range distances from each satellite to a terrain pixel on the ground. By observing the geometry of the Figure 2.3 and observing the positional relationship between the satellites and the terrain feature on the ground, we can derive the following:

$$\rho_1 = \sqrt{(H - \Delta z)^2 + (X + \Delta x)^2} \tag{2.4}$$

$$\rho_2 = \sqrt{(H - \Delta z + B\sin\alpha)^2 + (X + \Delta x - B\cos\alpha)^2} \tag{2.5}$$

When the SAR$_1$-transmitted signal reaches the ground, the signal gains a certain amount of phase, which it loses again when it travels the exact same distance back in the direction of SAR$_2$. Thus, we can define the interferometric path length δ, which refers to that change in displacement that the signal travels when it makes it pass from SAR$_1$ to SAR$_2$:

$$\delta = 2(\rho_1 - \rho_2). \tag{2.6}$$

The interferometric phase difference is just the phase equivalent of the interferometric path length, and thus can be easily defined as follows:

$$\phi = k\delta = \frac{2\pi}{\lambda}\delta = \frac{4\pi}{\lambda}(\rho_1 - \rho_2). \tag{2.7}$$

This phase difference would correspond to the ϕ_{int} in equation (2.3). In simulating our inteferogram for the purposes of this thesis, we will be creating a reasonable terrain in MATLAB, then going through the equations (2.4) through (2.7) to obtain the interferogram from the terrain. Once we have the interferogram on hand, we can go ahead with implementing the variety of techniques that we have developed in this thesis, including Projection and the range of unwrapping techniques, onto this interferogram to retrieve the original heights of the terrain that is being mapped.

2.3 The problem of wrapped phases

Now, we will investigate in great detail a problem with interferometry that we have described briefly in previous sections, and provide insight into potential solutions for it. This problem is the issue of wrapped phases. The absolute phases that we calculate in equation (2.7) are naturally continuously; since ρ_1 and ρ_2 are just straight line distances, there is no reason for equation (2.7) to be discontinuous. However, in reality, the phase differences measured are discontinuous, being confined in a range between $-\pi$ and π, and follows a strictly periodic cycle. In other words, the phases obtained from SAR systems are the absolute phases modulo 2π. The value of the quotient after this wrapping procedure is forever lost, and there is really no way to be perfectly sure what the actual phase value after phase wrapping has occurred.

One of the weakest and most commonly used methods to deal with this problem is to do the following: scan the interferogram in a raster fashion, and whenever you detect a 2π phase jump, add 2π back onto that particular interferogram pixel. This is similar to the *unwrap* function provided by MATLAB. While intuitive, this technique is handicapped by the lack of robustness to noise. Background noise is inevitable in any SAR system, and this may come from a variety of sources --- thermal radiation, imperfection of receiving and transmitting antennas, random distribution of input parameters, etc. The presence of noise means that it is often difficult to be confident that adding 2π to a pixel will perform the correct unwrap; in fact, it might be adding even more systematic noise into the system. As such, we will seek to implement other more effective phase unwrapping algorithms and compare their results. The *unwrap* algorithm that is available in MATLAB will not be used for any phase unwrapping comparisons in this thesis, since almost any other algorithm will outperform it, except in the cases of extremely low noise levels.

The 2 most commonly used phase-unwrapping algorithms used in literature today among many others (see [1]—[12] of Section 8) are the Weighted Least Square and the Branch Cut algorithms. Each of them has their own pros and cons. We will implement both for the purposes of this thesis, and perform a full-length comparison between the two to see which has a better noise-tolerance, since that is our primary concern when it comes to phase unwrapping. These will be investigated in the next chapter.

2.4 Retrieving height from the interferogram

After we successfully unwrap the phases from the interferogram, the next step would be to obtain the height of the terrain. We can start by rearranging equation (2.7) to get the following:

$$\delta = \frac{\lambda}{4\pi}\varphi \tag{2.8}$$

By simple geometric reasoning of Figure 2.3, we can obtain a relationship between the look angle θ and δ as follows:

$$\theta = \alpha + \arcsin\left(\frac{\delta}{B} + \frac{B}{2\rho_1} - \frac{\delta^2}{2B\rho_1}\right) \tag{2.9}$$

The height at a certain pixel, Δz, is thus given by:

$$\Delta z = H - \rho_1 \cos\theta \tag{2.10}$$

As such, given that we have the interferogram and all the necessary parameters (e.g. H, ρ_1, α , etc), we can obtain the entire digital elevation map from the phase map.

One caveat of retrieving the digital elevation map from the interferogram is the phenomenon of foreshortening. Foreshortening refers to fact that because slant ranges are usually the ones which are measured to be regularly spaced, the actual horizontal range intervals Δx are cannot be regularly spaced as long as the terrain is not a purely flat, horizontal surface. This effect happens primarily because the length of time that has passed from the time when pulses are emitted to the time when they return to the SAR depends on what they hit. Pulses that hit taller objects will return to the SAR system earlier, and pulses that hit a flat ground will return to the SAR system later than it otherwise would have. This effect, if not corrected for, creates the illusion that taller objects are nearer than they really should be, and shorter objects are further. We will not be going into detail in foreshortening; instead, we will simply assume that foreshortening has been corrected for automatically for us. This assumption is neither realistic nor representative of actual workings of a SAR system, but since SAR is a fairly modular system, that would not obstruct us in developing our phase unwrapping and projection algorithms.

3. Single Baseline Phase Unwrapping

3.1 The 2 satellite configuration

We now briefly look at the phase unwrapping process with 2 satellites (1 baseline), since some of the 3 satellite approaches are simply taking averages of phase unwrapping with 2 satellites. To do so, we first have to simulate the actual height terrain in MATLAB, as well as the system parameters, e.g. wavelength, noise level, and baseline separation, etc. We can then "measure" the phases that the satellites observe, and generate the interferogram from there. With this phase map, we introduce the Unweighted Least Squares (ULS) unwrapping method, the Weighted Least Squares (WLS) unwrapping method, and the Branch Cut Algorithm, and compare the results.

We start with a 1-dimensional view of the terrain, and seek to obtain the phase difference between the 2 satellites. At every azimuthal (cross-range) pixel interval, a shot of all the pixels in the range direction is taken, and stitching together all the cross range pixels generates a 2-dimensional phase map of the terrain. We assume we only know (or can measure) the following system parameters: baseline length B, baseline slant angle α, slant range of the first satellite ρ_1, wavelength of the signals transmitted received λ, and the distance of the first satellite from the ground, H. We are able to find (by transmitting and receiving the pulses) the wrapped phases of SAR 1 and 2. Figure 3.1 (which is identical to Figure 2.3) shows the SAR setup, as well as the corresponding system parameters.

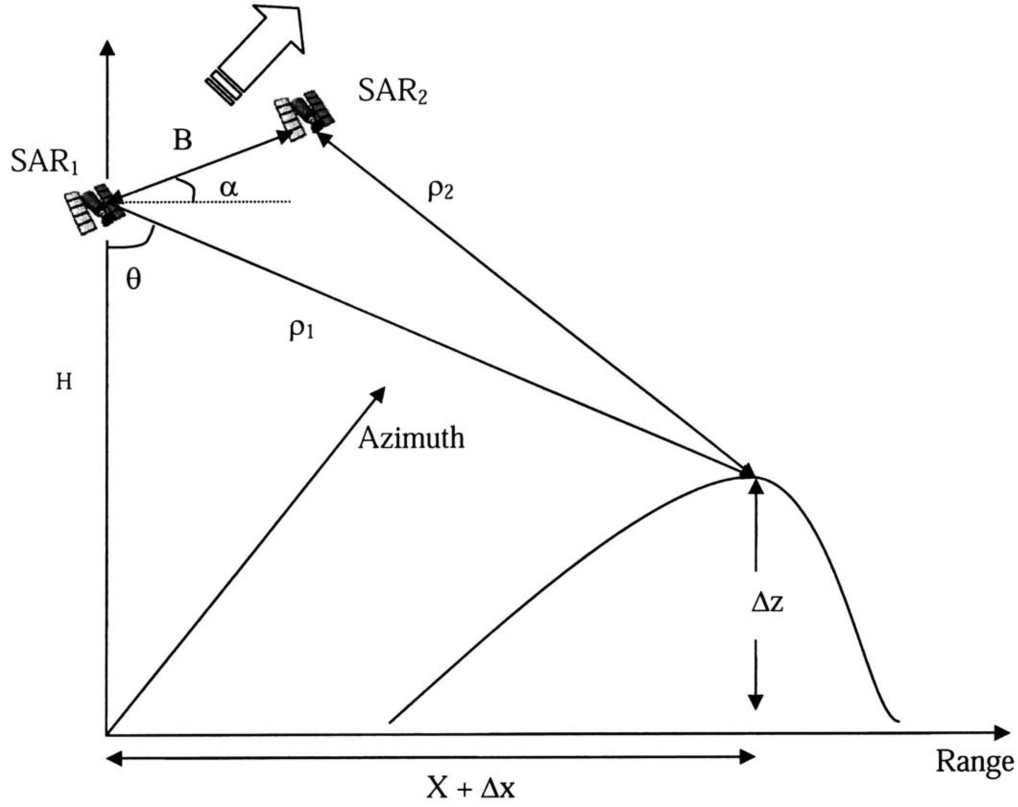


Figure 3.1: InSAR configuration for 2 satellites.

By simple geometry and reasoning, it can be easily shown that the interferometric phase difference, ϕ_{12}, is given by the following relationship:

$$\phi_{12} \equiv \phi_1 - \phi_2 = \frac{4\pi}{\lambda}(\rho_1 - \rho_2) \tag{3.1}$$

However, as we discussed before, the observed phase difference is wrapped modulo 2π, so the wrapped phase that we observe, ψ_{12}, is really given by

$$\psi_{12} \equiv W\langle\phi_{12}\rangle = \phi_{12} + 2\pi k, \quad k \in Integers \tag{3.2}$$

where the W operator maps every ϕ_{12} point to the range $(-\pi, \pi]$.

One quick way to potentially resolve the wrapped phase ambiguity is to just add 2π whenever we see a phase jump of more than π radians, since that probably corresponds to a wrapping of phases. This is akin to the *unwrap* function in MATLAB. However, this technique is seriously limited by the presence of noise, which is inevitably in all real systems. This unintelligent way of unwrapping will not detect pixels which suffer from abnormally high noise, or worse, shears in the terrain, and unwrap in a wrong fashion and lead to inaccurate height estimates. As such, we do not even discuss this method in any of our simulations or comparisons, since almost any other algorithm will outperform this unwrapping technique.

3.2 The noise model

Before we continue, we have to establish the noise model to be used so that we are all on the same page. We have picked a white noise model, with each pixel a random amount of noise, independent of other pixels. Thus, every pixel gets a random amount of phase noise ϕ_n (in radians) as follows:

$$\phi_n = n \cdot (2 \cdot rand(1) - 1) \cdot \frac{\pi}{180} \tag{3.3}$$

where rand(1) is a random number between 0 and 1 (similar to the MATLAB function *rand*), and n is the maximum amount of noise (in absolute value) each pixel can get. Thus, if we say that this interferogram has 40 degrees of noise, it really means each pixel can randomly have a phase noise of anywhere in between -40 degrees and 40 degrees.

To facilitate the inclusion of the notion of signal-to-noise ratio (SNR), we use the following definition of SNR (which is similar to that used in [14]):

$$SNR\ [dB] = 20\log\left(\frac{180}{\phi_n \cdot \pi}\right) \tag{3.4}$$

In reality, white noise may not be the best way to model noisy systems. There are many other noise models that we could have adopted, with Gaussian noise perhaps the most popular (and most commonly seen in nature) of the lot. Nevertheless, since the noise added is random anyways, and unless otherwise stated, the algorithms do not perform favorably for one noise model and not for another, the analyses should hold whichever noise model we choose to adopt, and one algorithm which works better for one noise model will intuitively also work better for another one.

3.3 The Unweighted Least Square unwrapping algorithm [1]

In [1], the Unweighted Least Square unwrapping method is introduced as a means to make use of gradients (changes in phases in the range and cross range axes) to produce more accurate unwrapped phases. In addition, we define the following parameters:

$$\Delta^{x}{}_{i,j} = W\left\langle \psi_{i+1,j} - \psi_{i,j} \right\rangle \tag{3.5}$$

$$\Delta^{y}{}_{i,j} = W\left\langle \psi_{i,j+1} - \psi_{i,j} \right\rangle \tag{3.6}$$

where i and j refer to differences in the range and cross-range axes respectively. From this point on, we'll just refer to i's and j's instead of range and cross-range, since we will be talking about matrices of phases, but there really is no difference between the two.

The least-squares solution requires, as the name implies, the $\phi_{i,j}$ that minimizes the square of the difference between the wrapped phase difference and the unwrapped phase difference. In mathematical form, we want to find the $\phi_{i,j}$ such that:

$$\sum_{i=0}^{M-2}\sum_{j=0}^{N-1}\left(\phi_{i+1,j} - \phi_{i,j} - \Delta^{x}_{i,j}\right)^2 + \sum_{i=0}^{M-1}\sum_{j=0}^{N-2}\left(\phi_{i,j+1} - \phi_{i,j} - \Delta^{y}_{i,j}\right)^2$$

is minimized. M and N represent the total number of pixels in the i and j directions.

An earlier paper by Hunt [4] has already shown that this problem can be solved in closed form. The unwrapped phase differences and the wrapped phase differences are related by the discrete Poisson's equation if the least squares constraint is to be satisfied and the 2D Discrete Cosine Transform (DCT) does the job perfectly.

The Unweighted Least Squares algorithm can be summarized succinctly below:

1. Let $\rho_{i,j}$ be the following:

$$\rho_{i,j} = \left(\Delta^{x}_{i,j} - \Delta^{x}_{i-1,j}\right) + \left(\Delta^{y}_{i,j} - \Delta^{y}_{i,j-1}\right) \tag{3.7}$$

Perform the 2D forward DCT on the array $\rho_{i,j}$ to yield the values $\rho_{hat,i,j}$.

2. Compute $\phi_{hat,i,j}$ according to the following equation:

$$\phi_{hat,i,j} = \frac{\rho_{hat,i,j}}{2\left(\cos\frac{\pi i}{M} + \cos\frac{\pi j}{N} - 2\right)} \tag{3.8}$$

3. Perform the 2D inverse DCT on $\phi_{hat,i,j}$ to obtain $\phi_{i,j}$, which are the least-squares unwrapped phase values that we seek.

3.4 Residues and the Branch Cut algorithm [11]

In [11], Goldstein et al talked about using "residues" to determine phase errors in interferograms. When we move from one pixel to the next, the change is designated a +1 if there is a change of more than $+\pi$ (half a cycle), a -1 if there is a change of more than $-\pi$, and a 0 otherwise. If there were no noise in the system, then going clockwise or counter-clockwise back to the same point that we started off in any fashion while summing up all the +1's, -1's, and 0's should always give us 0 in the end. This is analogous to walking from the top of a mountain down the hill, then go back up to the same point on the mountain via any other path – we get back to the same point. However, if there is noise in the system, then the phases will get randomly shifted up and down – and the phase at one pixel may be so large that it gets a +1 or a -1 relative to its neighboring pixels. Thus, if we integrate clockwise (or counter-clockwise) around, 4 pixels, we may get a residue – a +1 or -1. The presence of a residue thus tells us that there is inconsistency. Note, however, that the absence of a residue does not indicate the absence of noise. If the noise level is not high enough, the phases may get disrupted but not so much as to cause loss of consistency.

To illustrate the idea of residues, consider the following matrix of phases:

$$\begin{matrix} 0.2 & 0.8 \\ 0.4 & 0.6 \end{matrix}$$

The numbers represent the phases that are measured, normalized by 2π .If we integrate around the phases, we get a residue of +1 – an indication of inconsistency.

The next step of the unwrapping process is to connect nearby +1 and -1 residues with "cuts". These cuts act as walls so that integration paths can cross them, and so no net residues can be encircled by these residue paths. As such, no global errors can be generated, although local errors in the immediate neighborhood of the residues can still occur. Cuts are chosen such that the total length of them is kept to a minimum to minimize the total phase discontinuities in the interferogram.

When there are only a few residues around, the location of the optimum cuts is there for all to see. However, once more noise creeps in and noise levels shoot up, it would be computationally intensive to solve for the best cuts. The following algorithm is thus proposed:

1. Scan interferogram until a residue is found.

2. Place a 3 by 3 box around the residue, and continue scanning for another residue. If another residue is found, place a cut between them.

3. Designate the cut as "uncharged" if the residue is of the opposite sign, and repeat step 1. If residue is of the same sign as the original, move the box to the new residue and continue searching until either:

   a. an opposite residue is located and the resulting total cut is uncharged, or
   b. no new residues can be found.

4. If step 3b occurs, increase size of box by 2, and repeat algorithm from current starting residue.

Goldstein *et al* suggest introducing such "branch cuts" to surround areas on the interferograms with residues. Essentially, "branch cuts" are "walls" set up to prevent pixels near areas with residues from unwrapping using information from these phases, thus preventing the contaminated phases where the residues lie from spreading globally throughout the unwrapped data. However, this method fails when the noise level is too high (hence too many residues), since there is no clear way to determine which is the best path to take to unwrap around the residues. Worse, there may be so many residues in the phase map that no suitable branch cuts exist, resulting in isolated patches of the interferogram which are not processed by the algorithm. However, this does not automatically condemn the Branch Cut algorithm to insignificance. Instead, we can use the residues to determine the weights of the Weighted Least Squares algorithm, which is explained next.

3.5 The Weighted Least Square unwrapping algorithm [1],[24]

The Weighted Least Square algorithm is a critical extension to the Unweighted Least Square algorithm, since it allows us to give higher weights to phase points which we have more confidence in, and lower weights to those which we doubt are correct.

According to Ghalia and Romero [1], the Unweighted Least Square algorithm can be expressed in the following matrix form:

$$\overline{\overline{P}}\cdot\overline{\Phi} = \overline{\rho} \tag{3.9}$$

where $\overline{\overline{P}}$ is a matrix that performs the discrete Laplacian operator on a vector of the unwrapped phases $\overline{\Phi}$ (which we are trying to solve for), and $\overline{\rho}$ is a vector containing the discrete Laplacian operation on the wrapped phase differences, just like equation (3.7).

We have already established that equation (3.9) can be easily solved by the forward and inverse discrete cosine transforms.

However, if we know that certain wrapped phase values are corrupted by some degrading influence like noise, we can assign those phases less or even no weights so as to increase the robustness of the unwrapping process [24]. One way to know that wrapped phase values are indeed corrupted is to check for the presence of residues. If a residue is present, it means that the 4 neighboring pixels are inconsistent with one another. There is no sure-fire way to tell which pixel(s) are actually "wrong" – the positioning of the large phase jump is relative depending on the position of the 2π interval (in our case, the 2π interval is, as stated before, $(-\pi, \pi]$). Hence, whenever there is a residue, we assign zero weights to the 4 corresponding pixels, and if there is no (zero) residue, the weight stays at unity.

Unfortunately, the presence of the weighting matrix means that there is no closed form solution for the least squares problem, unlike in the unweighted case when the relationship between wrapped phase differences and unwrapped phase differences reduce to a discrete Poisson's equation. Ghalia and Romero [1] showed that incorporating iterative techniques that utilize our ability to solve the Unweighted Least Square problem into the algorithm are able to converge to the correct phase values in relatively little time.

3.6 Denoising ("Cleaning") [25]

Denoising (or "Cleaning") is the process of reducing the effects of noise on the phase values. Since noise is the primary cause of errors in phases and bad performance of phase unwrapping algorithms, denoising becomes a critical part of the phase unwrapping process, as well as the subsequent height retrieval step.

Since the noise in all the interferograms is uncorrelated white noise, averaging is perhaps the single most effective way to kill noise. Denoising, as explained in [25], is performed as follows:

1. The interferogram is expressed as a matrix of complex numbers of magnitude 1, $e^{j\psi_{i,j}}$, with the angle $\psi_{i,j}$ being the phase value of the corresponding pixel in the interferogram.
2. The denoised phase value at pixel (*i,j*) is then:

$$\psi_{denoised,i,j} = \frac{angle\left(\sum_{i-1}^{i+1}\sum_{j-1}^{j+1} e^{j\psi_{i,j}}\right)}{\left|\sum_{i-1}^{i+1}\sum_{j-1}^{j+1} e^{j\psi_{i,j}}\right|} = \frac{1}{9} angle\left(\sum_{i-1}^{i+1}\sum_{j-1}^{j+1} e^{j\psi_{i,j}}\right) \tag{3.10}$$

Essentially, to denoise pixel (i,j), the 9 complex numbers (pixel (i,j) itself and its 8 immediate neighbors) coding the phase information are summed up and averaged, then the phase value of that resultant complex number is taken to be the new phase value of (i,j). Intuitively, if the system is perfectly (and unrealistically) noiseless, then the denoising process averages the entire signal and none of the noise, so we would expect denoising to worsen things. However, this method should produce more accurate results if there is some noise in the system, since averaging across 9 pixels can filter out much of the noise.

One thing to note is that arithmetic averaging in the complex plane will produce good results only if the points to be averaged lie close together. Averaging $e^{j\frac{\pi}{2}}$ and $e^{-j\frac{\pi}{2}}$ will produce 0 – hardly a satisfactory result if the correct value is closer to $e^{j\frac{\pi}{2}}$. As such, we can see that complex averaging may give us results that are worse than that without denoising if the noise level gets too high or the intervals between neighboring pixels get too large, as can be the case if the terrain is changing too steeply, like when we're mapping a cliff.

3.7 Comparing Unweighted Least Square, Weighted Least Square, Branch Cut with denoising, and Weighted Least Square with denoising

We now compare the Unweighted Least Square, Weighted Least Square, and Branch Cut algorithms, as well as investigate the effects of denoising on the unwrapped results. We use the following table of parameters to perform simulations:

Parameter	Value used
B (m)	200
α (degrees)	35
Δx (m)	5
X (m)	$3 \cdot 10^5$
H (m)	$5 \cdot 10^5$
Δy (m)	5
Y (m)	$3 \cdot 10^5$
λ (m)	0.03 (X band)

Table 3.1: Parameters used for single baseline case.

[13] shows that vertical misalignment of the curves happens and is to be expected because the amplitude information is thrown away when doing interferometry. Since we are not concerned with the alignment problem in this thesis, it suffices to say that we will just align the mean values of the plots to be compared to that of the actual terrain used so that the error comparison can be meaningful. This mean value is thus given by:

$$\frac{1}{N}\sum_{k=1}^{N} h(x_k) \tag{3.11}$$

where N is the number of pixels and $h(x_k)$ is the height value at the pixel corresponding to the x_k range value.

The primary metric used for comparison will be mean root-mean-square (RMS) errors. For each noise level (as is defined in equation (3.3)), we run the simulation a fixed number of times, say 10. Since noise is introduced into the system, there will be a difference in unwrapped phases and the actual phases, and so there will be a difference between the retrieved heights and the actual terrain heights. This difference can be quantified in terms of the RMS error, which is given by:

$$h_{rms,i,j} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(h_{i,j} - h_{actual,i,j}\right)^2} \tag{3.12}$$

where N is the total number of times we run the simulation, $h_{i,j}$ is the retrieved height from the unwrapped phases, $h_{actual,i,j}$ is the actual terrain height that we know, and $h_{rms,i,j}$ is the RMS height error for that particular pixel. After we have the RMS height errors for all the pixels, we can count the number of pixels that fall into particular "bins" of error ranges, and thus plot a histogram to represent such results.

The following RMS histograms plot the distribution of errors. Clearly, the closer to zero the mean of the distribution, the better the phase unwrapping algorithm does. The best scenario should be the noiseless case, when the peak appears at the bin for zero error, with no spread along the horizontal axis. The performances of unweighted, weighted, weighted + cleaned, and branch cut + cleaned are investigated, and Figure 3.2 shows all four cases for a noise level of 40 degrees.

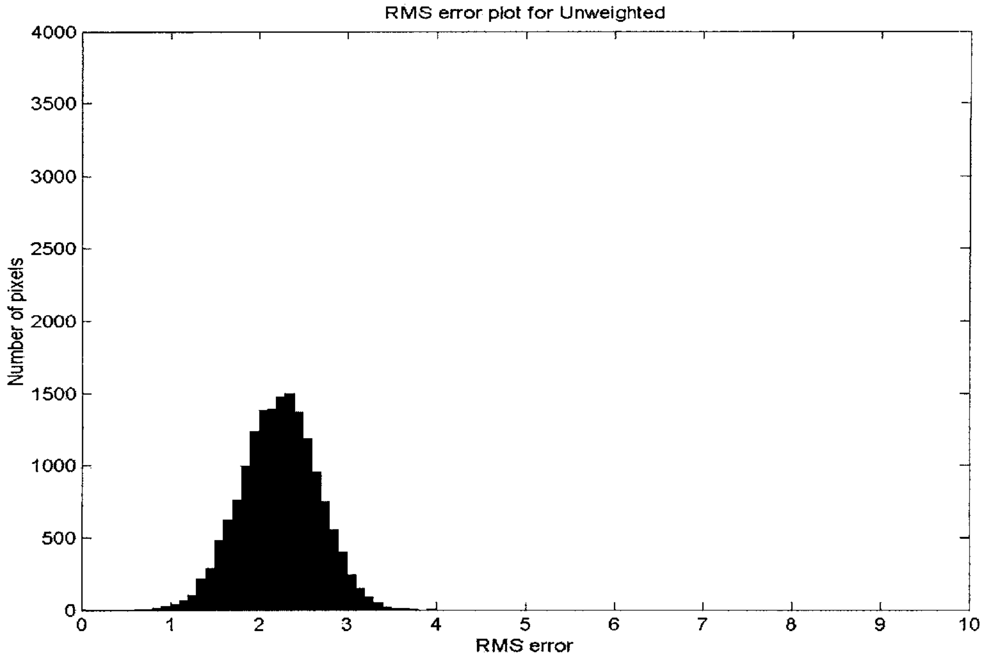


Figure 3.2a: RMS error plot for unweighted phase unwrapping, n = 40 degrees.

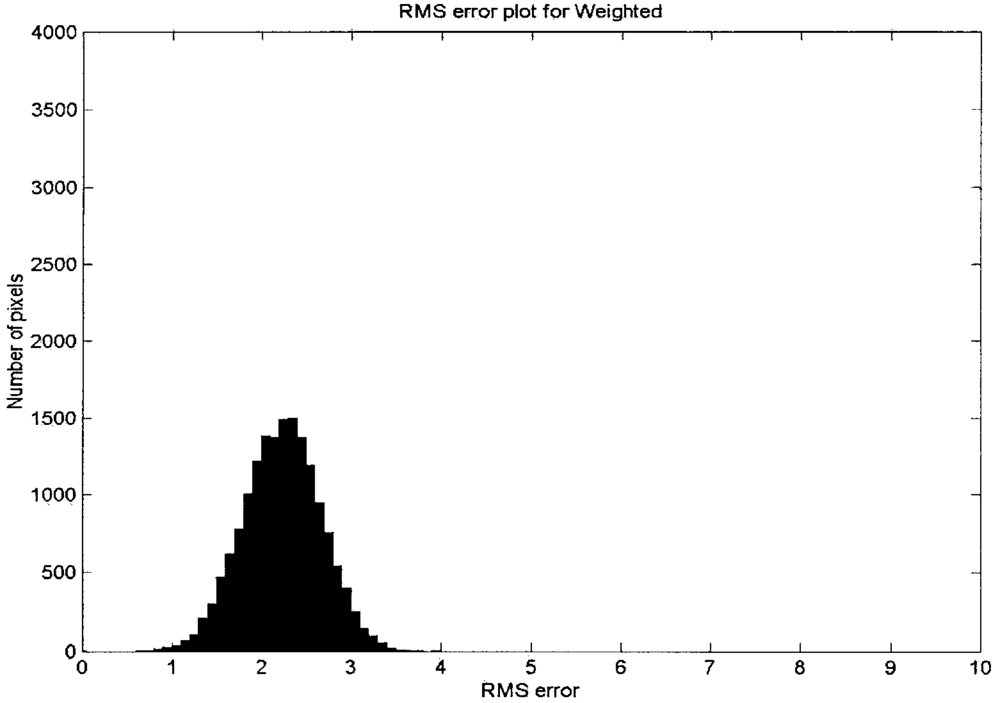


Figure 3.2b: RMS error plot for weighted phase unwrapping, n = 40degrees.

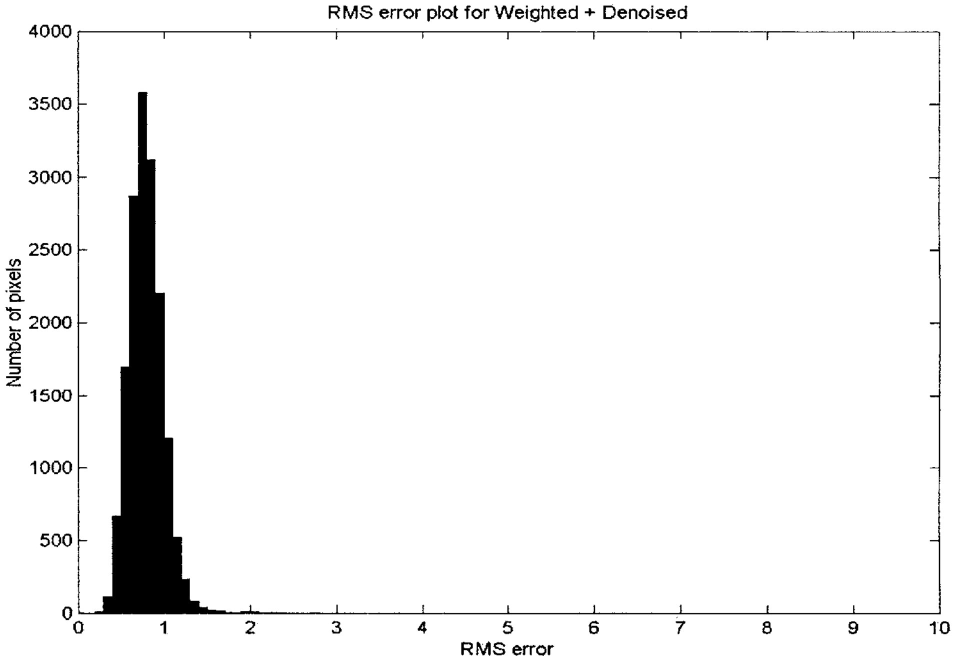


Figure 3.2c: RMS error plot for weighted phase unwrapping with denoising, n = 40 degrees.

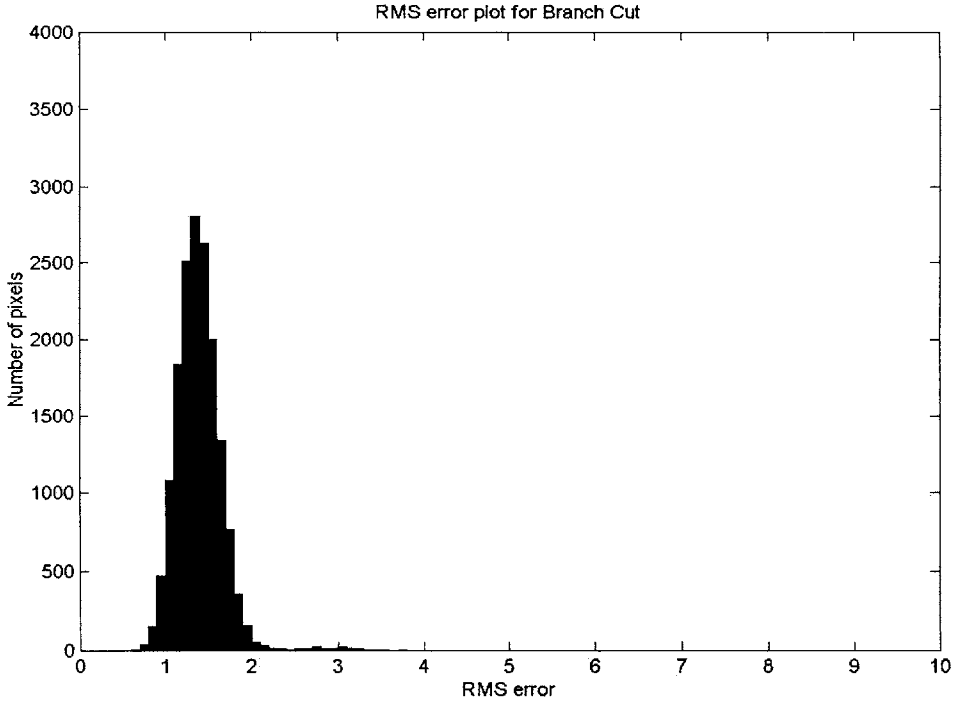


Figure 3.2d: RMS error plot for Branch Cut phase unwrapping with denoising, n = 40 degrees.

The mean of the distribution is the "balance point" of the distribution, and is defined to be:

$$h_{rms,mean} = \frac{\sum_{k} k \cdot N(k)}{\sum_{k} N(k)} \tag{3.13}$$

where the 2 summations refer to the number of bins used, *k* refers to the bin number, and *N(k)* refers to the number of pixels that fall into bin number *k*. This means that if most of the weight is centered on a particular RMS error bin, then the mean RMS height will naturally fall around that value too.

The RMS errors for varying noise levels for the three techniques are shown in Table 3.2, and a graphical representation of data in the table is shown in Figure 3.3. Note that "Residue Concentration" refers to the percentage of pixels that are determined to be affected by residues.

		Mean RMS Error (m)				Residue Concentration (%)
Noise (degrees)	SNR (dB)	Unweighted	Weighted	Weighted Cleaned	Branch Cut Cleaned	
0	∞	0	0	0.1095	0.2078	0
10	15.16	0.6875	0.6875	0.4954	0.5361	0
20	9.14	1.1753	1.1753	0.6404	0.6754	0
30	5.62	1.7055	1.7055	0.834	0.8637	0
40	3.12	2.2802	2.2429	1.1164	1.138	0.062
50	1.18	3.9661	2.7125	1.6039	1.603	3.05
60	-0.40	9.1408	3.2094	2.8651	6.2648	7.56
70	-1.74	16.489	6.0915	4.8518	8.5791	13.28
80	-2.90	23.8509	13.3311	8.4918	8.6203	18.96
90	-3.92	30.788	23.404	14.6903	8.7054	24.12

Table 3.2: Table of mean RMS error for varying noise levels.

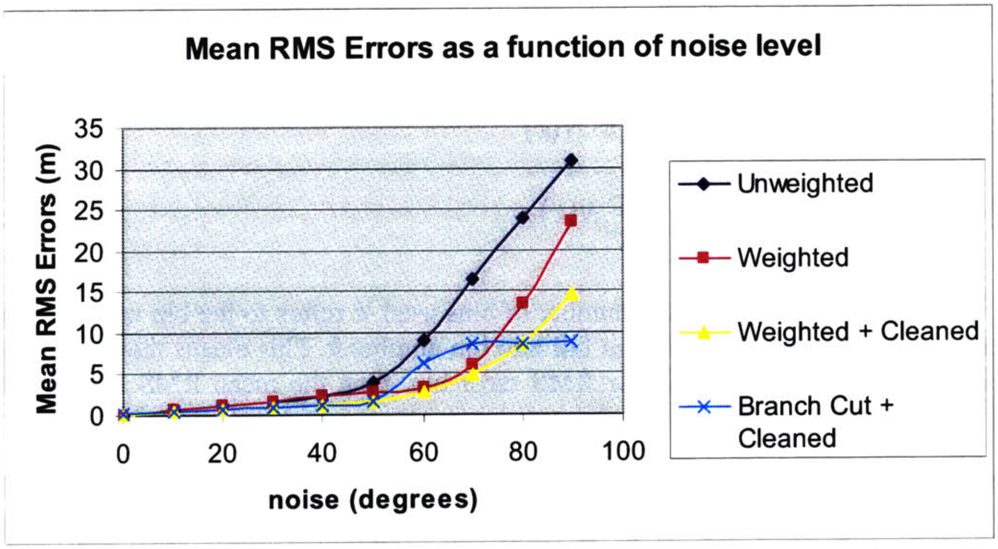


Figure 3.3: Plot of mean RMS error against noise level for Unweighted Least Square, Weighted Least Square, Weighted Least Square with denoising, and Branch Cut with denoising.

From Table 3.2 and Figure 3.3, we see that Weighted Least Square does at least as well as Unweighted Least Square in all noise levels, and does better whenever the residue concentration is a non-zero value. For example, at a SNR of 1.18dB (50 degrees noise), the mean RMS error for Weighting and denoising is only 60% that of just Weighted Least Square, and about 40% the error of Unweighted Least Square. This is a huge improvement. The Weighted Least Square method will have all weights of unity whenever there is no residue, which simply decomposes into the Unweighted Least Square problem, and when there is residue the weighting makes the phase unwrapping process more robust to noise. Denoising clearly helps to significantly reduce errors in the phases, since denoised phases do better in all noise levels except when there is no noise, which is unrealistic in any practical system. It is unclear whether denoising will help or hurt when noises get even higher than those presented here, but any noise floor higher is highly improbable and will bury the signal anyway so we expect that no phase unwrapping method will give results close enough to the terrain to be acceptable. The Branch Cut algorithm, coupled with denoising, does not perform as well as the weighted least square algorithms for SNR greater than -3dB (80 degree noise), and it is only when the noise SNR becomes an impractically small number like -4dB (90 degree noise) does the Branch Cut algorithm outperform the other algorithms. In fact, looking at Table 2, we find that the Branch Cut algorithm does worse than its Least Squares counterpart for all noise levels less than 90 degrees.

4. Multibaseline Phase Unwrapping – Iterative Dynamic Programming Approach with the Maximum a Posteriori Rule

There are many other phase unwrapping methods for multiple baselines that exist in literature today, and a few do not seem to have been as widely used as the others, although they seem to hold much promise in their mathematical approach to the problem. One of these which we would investigate in this section is the Dynamic Programming Approach using the MAP rule introduced by Ying, Frey, Koetter, and Munson in their papers [21] and [27]. These 2 papers describe the MAP technique being used in both single baseline and multiple baseline cases. We implement these algorithms in MATLAB, and then compare them with our existing techniques, namely Weighted Least Square and Branch Cut, to determine which method works best for our purposes.

4.1 The 3 satellite configuration

Wong's preliminary report [13] found that placing 3 satellites in a triangle formation will be able to give us access to many different ratios of baselines if the 3 satellites rotate around clockwise in a cartwheel fashion. In this section, we investigate the advantages of 3 baselines in the unwrapping process. Figure 4.1 below shows the setup of the 3 satellites, as well as the relevant parameters.

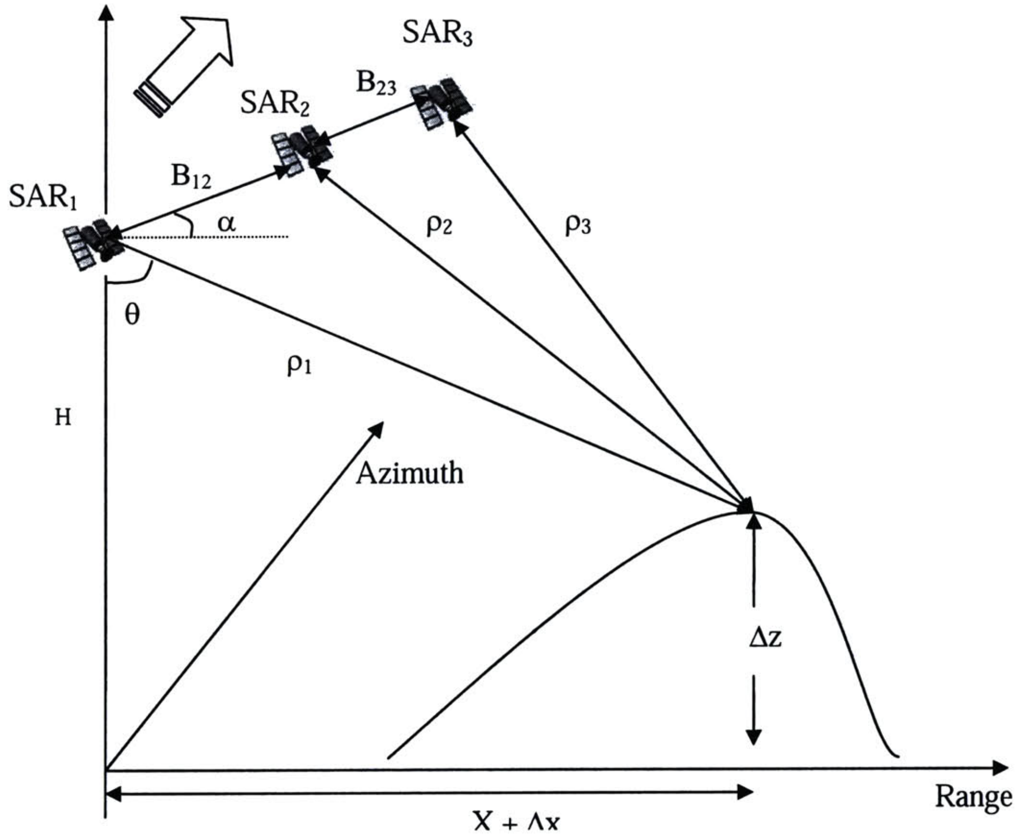


Figure 4.1: InSAR configuration for 3 satellites, collinear arrangement.

All parameters used will be kept as they were when we were dealing with only 2 satellites, with the exception of baseline lengths B_{12}, B_{23}, and B_{13}, which were given earlier in Table 3.1.

Parameter	Value used
B_{12} (m)	150
B_{23} (m)	50
B_{13} (m)	200

Table 4.1: Values of baseline lengths used for 3 baselines.

Note that $B_{13} = B_{23} + B_{12}$, as is required by geometry. B_{13} is the set to be the same length as B for 2 satellites, and so we would expect the analysis of the interferogram generated for satellite pairs SAR_1 and SAR_3 to yield similar results to that presented in the previous section.

4.2 Height averaging

Since the phases (and hence the retrieved heights) are affected by random noise, it is intuitive to expect that simple averaging of the retrieved heights will reduce the effects of noise even further. Wong et al [13] showed that averaging with 3 baselines work best if we only average the heights retrieved from the 2 longest baselines. That preliminary report derived an inversely linear relationship between RMS height error and baseline length, expressed as the following:

$$RMS\ Error \propto \frac{n}{B} \tag{4.1}$$

Wong's report showed that averaging through the 2 longest baselines gives the best results, whereas if we average through all 3 baselines, the large errors from the shortest baseline contaminates the result more than the averaging itself helps, so the retrieved height errors are worse.

Height averaging is a simple-minded and relatively effective technique to obtain good results regardless of whichever phase unwrapping algorithm we choose to employ. Thus, we will use the Weighted Least Square unwrapping algorithm in tandem with averaging as follows:

1. Unwrap phases ψ_{12} ,and ψ_{13} separately using the Weighted Least Square with denoising algorithm from the previous section. The unwrapped phases are ϕ_{12} and ϕ_{13}.

2. Retrieve the heights h_1 and h_2 respectively from ϕ_{12} and ϕ_{13}.
3. $h_{avg} = \frac{1}{2}(h_1 + h_2)$

4.3 Dynamic programming for single baseline

The primary assumption of this approach is that the unwrapped phases (and thus the terrain) are distributed according to a first-order Gaussian Markov random field (GMRF); in essence, this means that if we take the generation and distribution of phases as a stochastic process, then it must be both normally (or Gaussian) distributed, and that the state (phase) at any particular pixel only depended only on the previous state (a Markov process).

As expressed in equation (3.2), every wrapped phase $\psi_{i,j}$ has to have a corresponding integer $k_{i,j}$ added to it to produce the desired unwrapped phase $\phi_{i,j}$. The set of unknown $k_{i,j}$'s which we are trying to solve for is then denoted **k**. The maximum a posteriori (MAP) estimator then requires that **k** satisfy the following:

$$\mathbf{k} = \arg\min \sum_{i,j} \left[\left(\psi_{i,j} + k_{i,j} - \psi_{i-1,j} - k_{i-1,j} \right)^2 + \left(\psi_{i,j} + k_{i,j} - \psi_{i,j-1} - k_{i,j-1} \right)^2 \right]$$

This algorithm says that we should seek to find the k's such that differences of phases of adjacent pixels are minimized. This result makes sense intuitively since adjacent pixels are more likely to be closer together than further apart.

To find all the k's that minimize the above equation, one way would be to use brute force – assuming that all k's lie within the range [-K, K], search for every single one of the $(2K+1)^{MN}$ possibilities if the image size is of size M by N. Clearly, this method is not scalable. Any reasonably sized image would have a M and N whose product is prohibitively large. Instead, the following 2 changes are made:

1. $k_{i,j}$ is treated as intermediate shifts, an element from the set {-1, 0, 1}, and this restricted search is iterated K times to achieve essentially the same effect as the greedy algorithm since the phases can still expand to the range [-K, K]. The number of searches per iteration is thus reduced to 3^{MN} instead of $(2K+1)^{MN}$. This exponential algorithm is still not scalable with respect to image size, but at least this change increases the overall speed of the algorithm.

2. Instead of processing the whole interferogram at once, rows are processed W of them at a time. Dynamic programming is employed within these W rows via a "sliding window" to search for an optimal k, and the image is updated by adding k before moving on. A sliding window of W=2 is shown below:

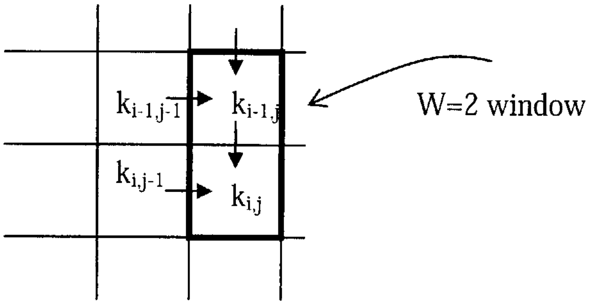


Figure 4.2: The sliding window (of size W=2) in the MAP technique.

For W=2, a state is defined as:

$$s(i,j)=\left(k_{i-1,j},k_{i,j}\right) \tag{4.2}$$

Assuming that the difference in k between adjacent pixels does not exceed 1, there are seven states that can exist in a dynamic program. The metric increment of any particular state $s(i,j)$ is then defined as:

$$\begin{aligned}\Delta d(i,j)=&\left(\psi_{i-1,j}+k_{i-1,j}-\psi_{i-1,j-1}-k_{i-1,j-1}\right)^2\\&+\left(\psi_{i,j}+k_{i,j}-\psi_{i-1,j}-k_{i-1,j}\right)^2\\&+\left(\psi_{i,j}+k_{i,j}-\psi_{i,j-1}-k_{i,j-1}\right)^2\\&+\left(\psi_{i-1,j}+k_{i-1,j}-\psi_{i-2,j}-k_{i-2,j}\right)^2\end{aligned} \tag{4.3}$$

and the accumulated metric of state $s(i,j)$ is:

$$d(i,j)=\min\left[d(i,j-1)+\Delta d(i,j)\right] \tag{4.4}$$

After scanning the W (2 in this case) rows, the unit increments of k values of all the pixels in the rows are determined by observing which combination of k values give the lowest accumulated cost metric. To reduce directional effects, we alternate the raster scan from iteration to iteration. In this way, after a maximum of K iterations, the wrapped phase map should be fully unwrapped. An added advantage of this approach is the spatial efficiency – memory used at each stage can be reused for the next stage.

4.4 Multibaseline InSAR elevation estimation via dynamic programming

The single baseline approach described above can be extended to the multibaseline domain. However, the main problem of extending to multibaseline is that the different interferograms have entirely different wrapping rates defined by their respective baselines, so processing their phases consistently in a single dynamic program becomes a challenge. Ying et al described a way to achieve this in their paper [27] – instead of trying to estimate the correct phases, it would be better to estimate the heights generated from the phases, which are consistent with one another as long as the inversion from phases to heights is done correctly. However, in the paper, their height estimation is expressed as a linear function of unwrapped phase as follows:

$$h(i,j) \approx a\left(\psi_{i,j} + 2\pi k_{i,j}\right) \tag{4.5}$$

where

$$a = \frac{\lambda\rho}{4\pi B} \tag{4.6}$$

with the parameters being the ones laid out in Figure 3.1. This equation is only approximately correct, because it assumes the following things:

1. The elevation angle, α , is set to be 90 degrees (so the satellites form a straight, vertical line). Clearly, changing α by even a slight bit will result in wholly different phases, and this assumption applies a constraint that might be unreasonable when we are using the satellites to map a terrain, since it is impossible to require the satellites to fly in a vertical line for the duration of the operation. However, for now, we will still use this assumption in our simulation to evaluate the performance of the algorithm.

2. $\frac{\lambda\phi}{4\pi B}$ is assumed to be much greater in magnitude than $\frac{B}{2\rho}$. This assumption has to hold for the retrieved height to be linearly proportional to the unwrapped phase. It is clear, however, that the following assumption is valid only if B is small, or ρ is large, which might not be the case for all practical scenarios.

For the remainder of the discussion, we assume the above 2 conditions are satisfied; also, for all our simulations utilizing, we set the parameters such that the assumptions hold. The multibaseline MAP technique then works in a similar fashion as the single baseline case, but instead of using the phase difference as the metric, we use the height difference as the metric $V_j(s_{j-1},s_j)$, as follows:

$$
\begin{aligned}
V_j\left(s_{j-1}, s_j\right) = & \frac{\left(h(i,j) - a_1\left(\psi_1(i,j) + 2\pi k_1(i,j)\right)\right)^2}{a_1^2 \sigma_n^2} \\
& + \frac{\left(h(i,j) - a_2\left(\psi_2(i,j) + 2\pi k_2(i,j)\right)\right)^2}{a_2^2 \sigma_n^2} \\
& + \frac{\left(h(i+1,j) - a_1\left(\psi_1(i+1,j) + 2\pi k_1(i+1,j)\right)\right)^2}{a_1^2 \sigma_n^2} \\
& + \frac{\left(h(i+1,j) - a_2\left(\psi_2(i+1,j) + 2\pi k_2(i+1,j)\right)\right)^2}{a_2^2 \sigma_n^2} \\
& + \frac{\left(h(i,j) - h(i+1,j)\right)^2}{\sigma_d^2} \\
& + \frac{\left(h(i,j) - h(i-1,j)\right)^2}{\sigma_d^2} \\
& + \frac{\left(h(i,j) - h(i,j-1)\right)^2}{\sigma_d^2} \\
& + \frac{\left(h(i+1,j) - h(i+1,j-1)\right)^2}{\sigma_d^2}
\end{aligned}
\tag{4.7}
$$

We need to express the continuous variables *h(i,j)* and *h(i+1,j)* in the equation (4.6) in terms of the discrete variables *k(i,j)* and *k(i+1,j)* by solving the following:

$$
\begin{aligned}
& \frac{\partial V}{\partial h(i,j)} = 0 \\
& \frac{\partial V}{\partial h(i+1,j)} = 0
\end{aligned}
\tag{4.8}
$$

for h(i,j) and h(i+1,j) and substituting into V. Solving the simultaneous differential equations yield the following h(i,j) and h(i+1,j):

$$
\begin{bmatrix} h(i,j) \\ h(i+1,j) \end{bmatrix} = \frac{1}{pq - s^2} \begin{bmatrix} cq + sd \\ cs + pd \end{bmatrix}
\tag{4.9}
$$

where

$$
p = \frac{1}{a_1^2 \sigma_n^2} + \frac{1}{a_2^2 \sigma_n^2} + \frac{3}{\sigma_d^2}
\tag{4.10a}
$$

$$
q = \frac{1}{a_1^2 \sigma_n^2} + \frac{1}{a_2^2 \sigma_n^2} + \frac{2}{\sigma_d^2}
\tag{4.10b}
$$

$$s = \frac{1}{\sigma_d^{\,2}} \tag{4.10c}$$

$$c = \frac{1}{a_1\sigma_n^{\,2}} + \frac{1}{a_2\sigma_n^{\,2}} + \frac{h(i,j-1)}{\sigma_d^{\,2}} + \frac{h(i-1,j)}{\sigma_d^{\,2}} \tag{4.10d}$$

$$d = \frac{1}{a_1\sigma_n^{\,2}} + \frac{1}{a_2\sigma_n^{\,2}} + \frac{h(i+1,j-1)}{\sigma_d^{\,2}} \tag{4.10e}$$

The algorithm proceeds in a fashion very similar to that presented in the single baseline case, but this time we are finding both **k**$_1$ and **k**$_2$, so the computational complexity is increased by a squared power. The results of the algorithms, when compared against the previous methods, are shown in the next section.

4.5 Comparing the various methods

Now, we are ready to compare the multibaseline phase unwrapping methods at our disposal. Since we have to satisfy the assumptions as stated in Section 4.4, we use the following table of parameters:

Parameter	Value used
B_{13} (m)	100
B_{12} (m)	70
B_{23} (m)	30
α (degrees)	90
X (m)	$3 \cdot 10^6$
Y (m)	$3 \cdot 10^6$

Table 4.2: Parameters used for MAP simulation.

All other parameters are assumed to be held the same as those in Table 3.1.

The 4 methods which will be compared are the following: Single baseline MAP with averaging, Multibaseline MAP, Weighted Least Square with averaging, and Branch Cut algorithm with averaging. The results are shown below:

Noise	SNR (dB)	Mean RMS Error (m)			
		Single Baseline MAP + Averaging	Multibaseline MAP	Weighted Least Square + Averaging	Branch Cut + Averaging
10	15.16	6.7987	5.4102	5.1718	5.5814
20	9.14	9.5987	7.9484	7.9983	8.5091
30	5.62	12.5371	11.0188	10.9426	11.3235
40	3.12	16.1309	14.6555	14.5754	14.9226
50	1.18	19.4803	18.125	17.9594	18.1592
60	-0.40	23.478	21.6344	21.6472	21.9309
70	-1.74	30.1622	25.6016	26.8802	27.0036
80	-2.90	39.0032	34.1117	33.2328	33.9233

Table 4.3: Results of different multibaseline unwrapping techniques for varying noise levels.

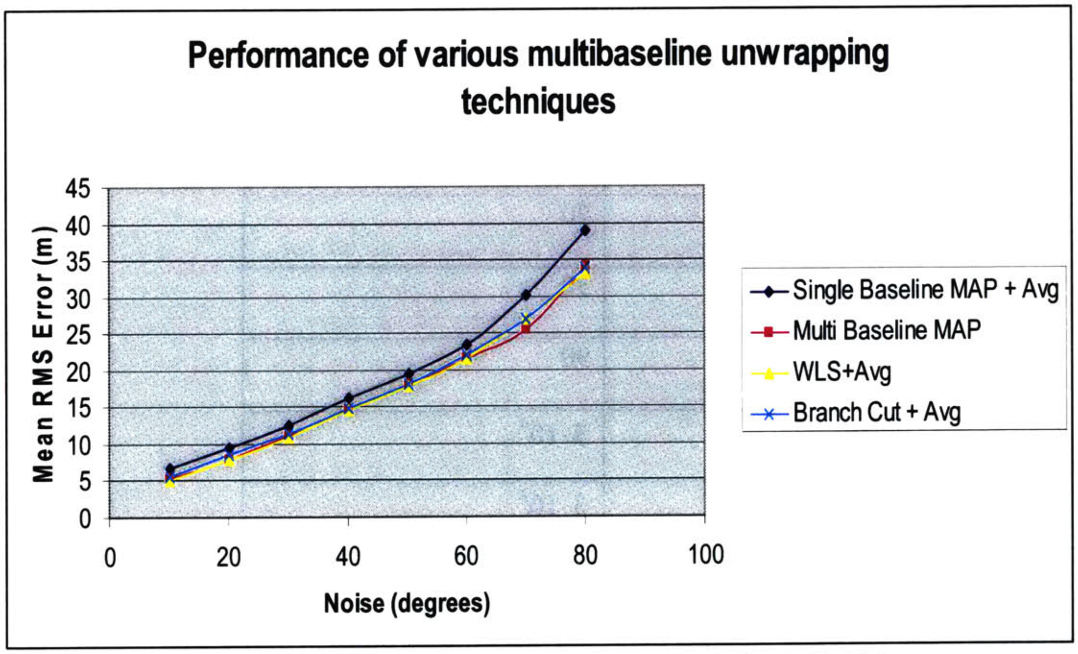


Figure 4.3: Graphical representation of results in Table 4.3.

The results from Table 4.3 and Figure 4.3 show that for a SNR ratio as low as -3dB, the Weighted Least Square method coupled with height averaging still outperforms the other algorithms, with multibaseline MAP coming in a close second for most of the noise levels. Although the mean height error difference is usually not significant (usually less than 1 meter between multibaseline MAP and Weighted Least Square with Averaging), it is still a win nonetheless for WLS with averaging. The single baseline MAP with averaging does the worst job in terms of height retrieval.

5. The 2-D and 3-D Projection Methods

5.1 Introduction to projection

In this section, we investigate a new noise-reduction technique when 3 satellites are present in a system. This new technique, called Projection, is a powerful tool that can potentially lead to increased robustness of the InSAR height retrieval process. In essence, projection relies on the very configuration of the satellites to deduce the behavior of the phases, and thus, give us the ability to classify any other behavior of the phases as the actions of noise. Because projection relies heavily on the physical configuration of satellites, the method inevitably will be jeopardized when the configuration itself is noisy or not ideal. Nonetheless, we press ahead to investigate how projection can help reduce noise in a multibaseline system.

5.2 The 3 satellite configuration revisited

Figure 5.1 below shows the 3 satellite configuration, and is identical to Figure 4.1.

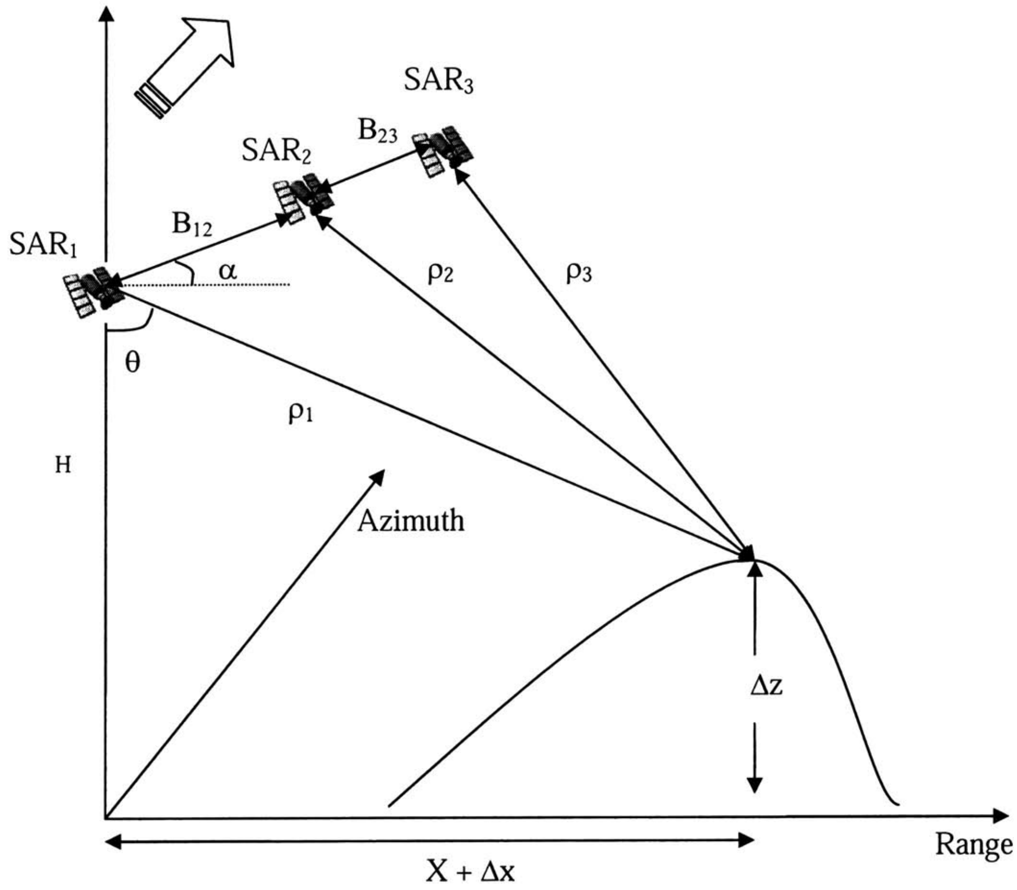


Figure 5.1: InSAR configuration for 3 satellites, collinear arrangement.

All parameters used will be kept as they were when we were dealing with only 2 satellites, with the exception of baseline lengths B_{12}, B_{23}, and B_{13}, which were given earlier in Table 3.2.

Parameter	Value used
B_{12} (m)	150
B_{23} (m)	50
B_{13} (m)	200

Table 5.1: Values of baseline lengths used for 3 baselines.

Note that $B_{13} = B_{23} + B_{12}$, as is required by geometry. B_{13} is the set to be the same length as B for 2 satellites, and so we would expect the analysis of the interferogram generated for satellite pairs SAR_1 and SAR_3 to yield similar results to that presented in the previous section. In particular, we would expect that

$$RMS\ Error \propto \frac{n}{B} \tag{5.1}$$

as Wong *et al* showed in their report [13].

5.3 Unambiguous Range Magnification (URM)

The URM method is a technique that employs the different rates of phase unwrapping due to the different baseline lengths to expand the range of unambiguous phases from 2π to a multiple of 2π. At its best, if all the actual phase values lie between 0 and this multiple of 2π, no unwrapping is necessary – all the phases are already "unwrapped" in the sense that they are already what they should be if unwrapping had actually taken place. At the very least, this expansion of the unambiguous range minimizes the amount of unwrapping work that needs to be done, which intuitively is a good thing – we don't want to unwrap if we don't have to, since the unwrapping algorithm chosen may introduce errors of its own.

The fundamentals of the URM stems from the following figure, which shows the close-up of the 3 antennas in Figure 5.1:

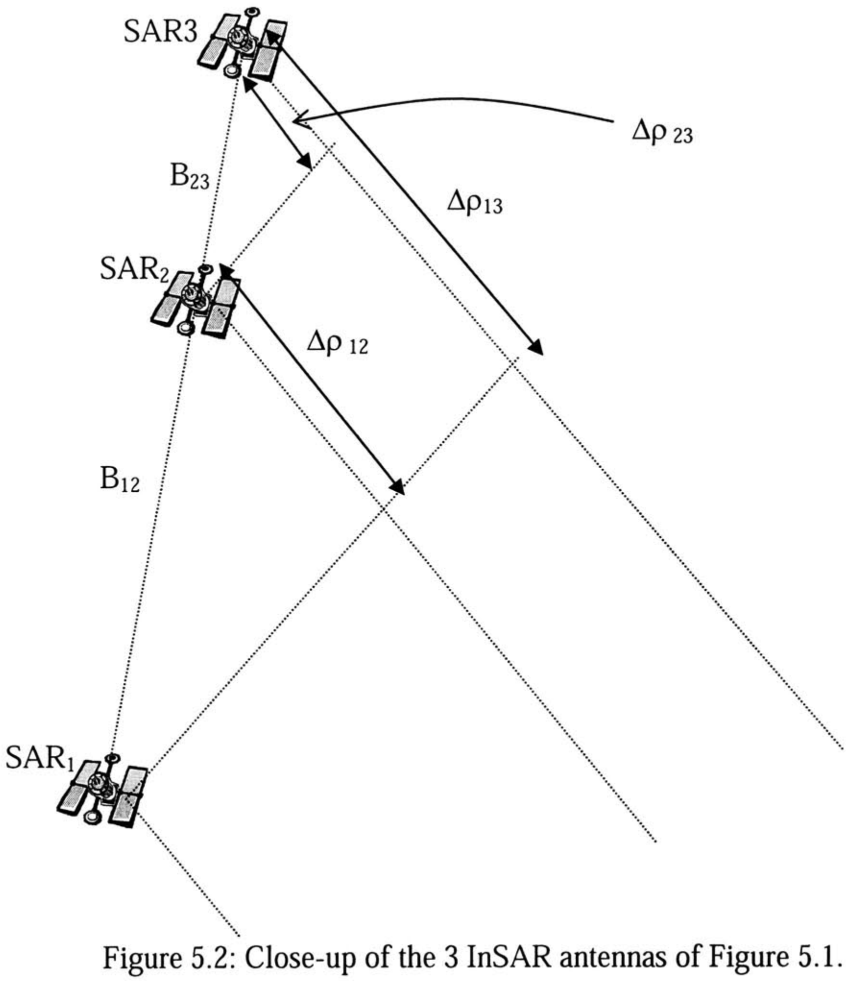


Figure 5.2: Close-up of the 3 InSAR antennas of Figure 5.1.

Since the slant ranges (ρ_1, ρ_2, and ρ_3) are on the order of 10^5 while the baselines (B_{12}, B_{23}, and B_{13}) are on the order of 10^2, a close-up of the 3 InSAR antennas will show the slant ranges vectors to be essentially parallel to one another. This in essence means there are 3 similar triangles that make up the geometry of the satellites, each identified by a baseline.

The actual phase values and slant range differences are related by the following simple equations:

$$\phi_{12} = \frac{4\pi}{\lambda}\Delta\rho_{12} \tag{5.2a}$$

$$\phi_{13} = \frac{4\pi}{\lambda}\Delta\rho_{13} \tag{5.2b}$$

$$\phi_{23} = \frac{4\pi}{\lambda}\Delta\rho_{23} \tag{5.2c}$$

Since phase values are always measured as wrapped values according to equation (3.2), equations (5.2a), (5.2b), and (5.2c) decompose into the following:

$$\psi_{12} + k_1 \cdot 2\pi = \frac{4\pi}{\lambda}\Delta\rho_{12} \tag{5.3a}$$

$$\psi_{13} + k_2 \cdot 2\pi = \frac{4\pi}{\lambda}\Delta\rho_{13} \tag{5.3b}$$

$$\psi_{23} + k_3 \cdot 2\pi = \frac{4\pi}{\lambda}\Delta\rho_{23} \tag{5.3c}$$

In addition, by the rule of similar triangles, the slant range differences and baselines have to be related by simple ratios:

$$\frac{B_{12}}{\Delta\rho_{12}} = \frac{B_{13}}{\Delta\rho_{13}} = \frac{B_{23}}{\Delta\rho_{23}} \tag{5.4}$$

Thus, we have the following relationships between wrapped phase values, which we measure, and the baseline lengths, which we know:

$$\frac{\psi_{13} + k_2 \cdot 2\pi}{\psi_{23} + k_3 \cdot 2\pi} = \frac{B_{13}}{B_{23}} \equiv URM_1 \tag{5.5a}$$

$$\frac{\psi_{12} + k_1 \cdot 2\pi}{\psi_{23} + k_3 \cdot 2\pi} = \frac{B_{12}}{B_{23}} \equiv URM_2 \tag{5.5b}$$

From equation (5.5a), it is clear that the wrapped phase ψ_{13} is a linear function of ψ_{23}, with the slope being URM_1. However, the vertical (ψ_{13}) intercepts, which depend on the k_2 and k_3 wrapping integers, are unknown and increment or decrement whenever there is a wrapping of the phase. Moreover, both ψ_{13} and ψ_{23} are wrapped values, so they are bound within a square box of length 2π. The exact same thing can be said for equation (5.5b), with the appropriate changes made to reflect the relationship between ψ_{12} and ψ_{23}.

This phenomenon is best explained in graphical terms, and that is shown in Figures 5.3 and 5.4 below. Since we use B_{12} = 150m and B_{23} = 50m (Table 5.1), URM_1 = 4, and URM_2 = 3. This is clearly seen by observing the slopes of the graphs of Figures 5.3 and 5.4.

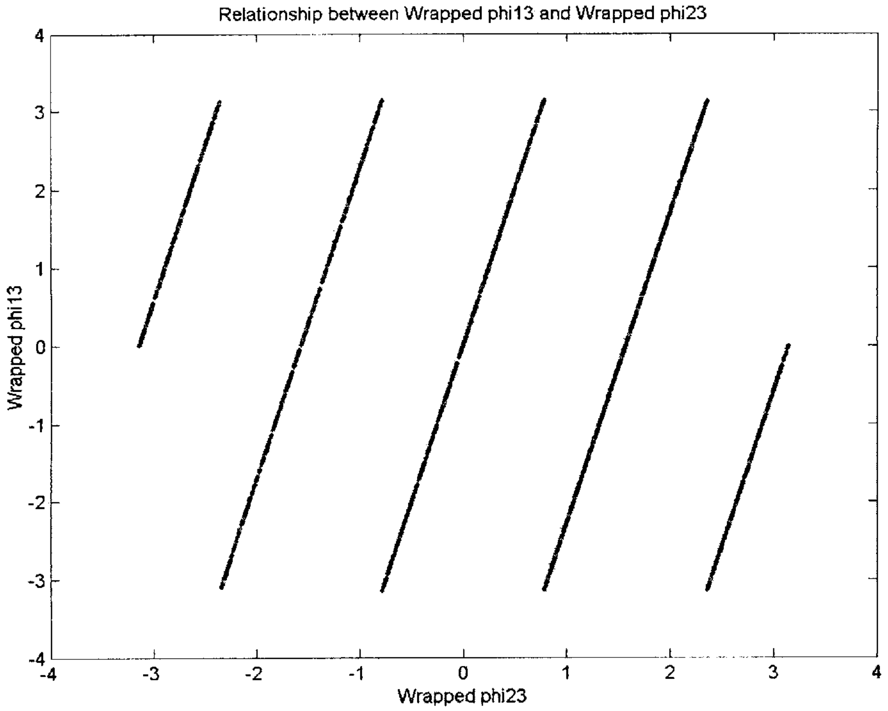


Figure 5.3: Relationship between ψ_{13} and ψ_{23}.

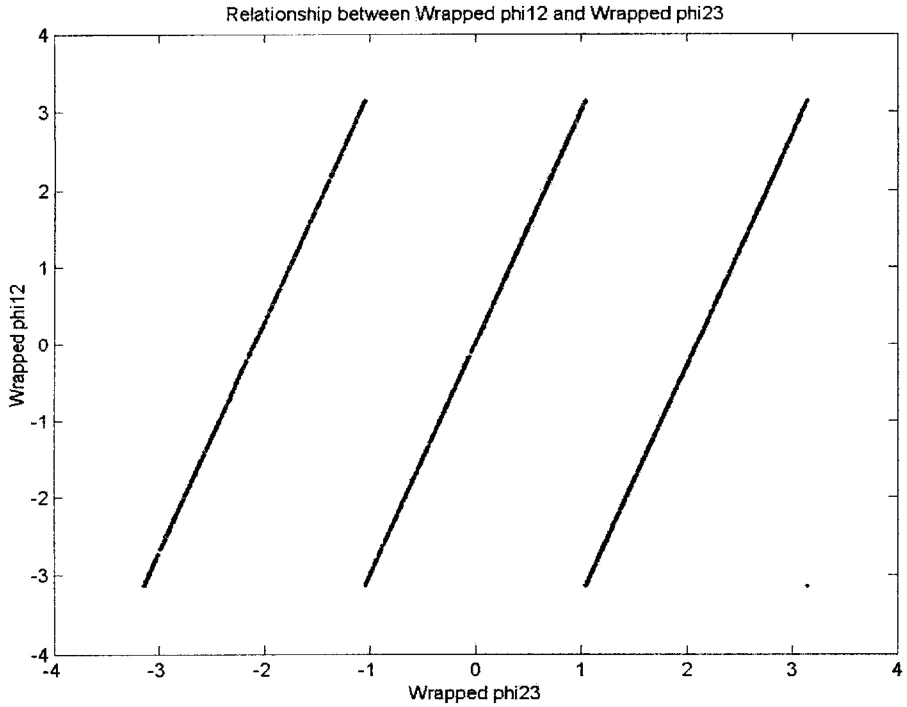


Figure 5.4: Relationship between ψ_{12} and ψ_{23}.

Xu et al [13] were the first to propose plotting these wrapped phases in this manner, and it is known that as long as URM is a rational number, the number of line segments will be finite, i.e. they will reconnect in a wrapped sense when they hit the edges of the box. Clearly, a (ψ_{13}, ψ_{23}) couple for any pixel (i,j) will now uniquely define the phase information for that pixel within a much larger range, since it is now in the ψ_{13}-ψ_{23} plane. Alternatively, we can link the line segments together without losing the increased range, and this means we increase the unambiguous range of one or both of the axes by an overall URM times.

Figure 5.5 shows the result of linking the line segments of Figure 5.3. In this case, ψ_{13} expands 4-fold to encompass a range from 0 to 8π (25.13) radians.

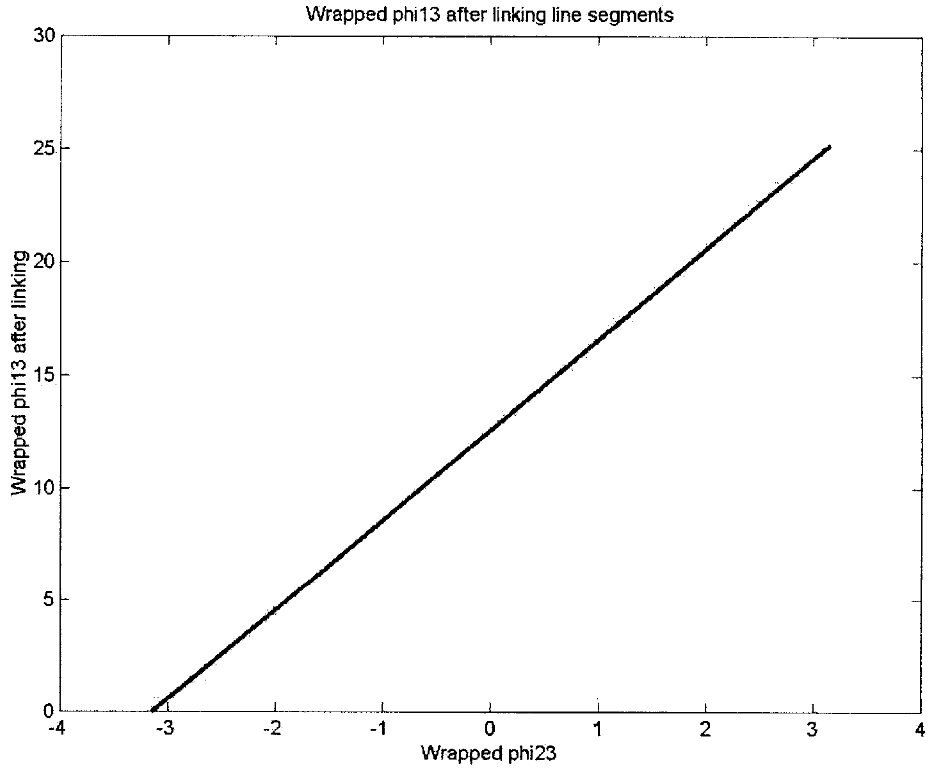
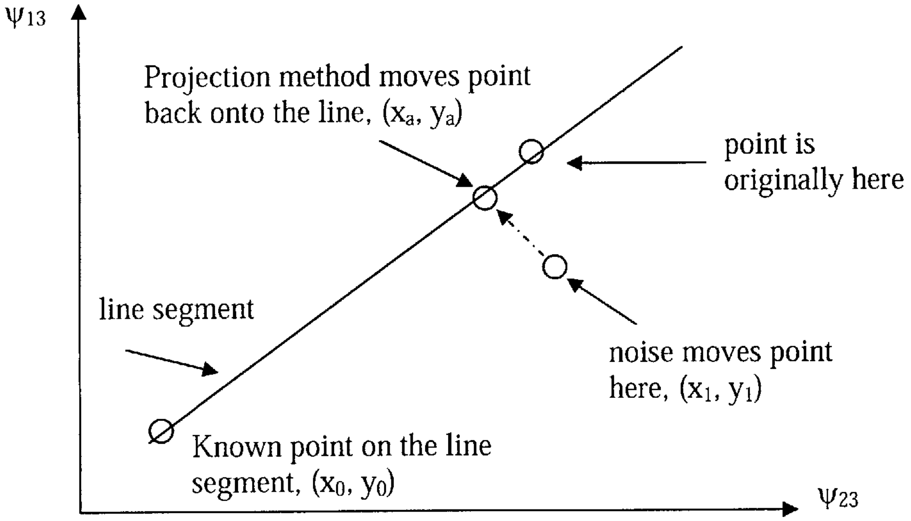


Figure 5.5: Figure 5.3 after linking line segments together.

5.4 Projection method

Theoretically, the URM technique allows us to expand the unambiguous range to any arbitrary size – just increase the ratio of B_{13} to B_{23}. However, when there is noise in the system, points will get shifted away from the line segments. Points that do not lie on a line segment have to be inconsistent with equations (5.5a) and (5.5b), which are derived purely from the geometry of the satellite configuration. This is thus a clear indication that noise has disrupted the real values of the wrapped phases. Xu et al [13] suggested that the best way to resolve this problem is to use the "Projection Method" – whenever a point lies away from a line segment, we place it back onto the nearest line segment with the shortest Euclidean distance. This ensures that the phase values maintain consistency of the geometry of the 3-satellite configuration, and also allows us to place the phase points closer (or even onto) the actual wrapped points. Figure 5.6 shows this process in action.



ψ13
Projection method moves point back onto the line, (xa, ya)
point is originally here
line segment
noise moves point here, (x1, y1)
Known point on the line segment, (x0, y0)
ψ23

Figure 5.6: Illustration of the projection method.

Now, we try to frame the 2-D Projection in a solid, mathematical framework. Referring to Figure 5.6, we name a few key points on the diagram by the following names:

1. a known point on the line segment is defined as (x_0, y_0)
2. an actual ψ_{23} - ψ_{13} phase couplet defined by the interferogram is defined as (x_1, y_1)
3. The point that we want to find, i.e. the point on the line segment that is of the shortest Euclidean distance to (x_1, y_1) is defined as (x_a, y_a).

With that, we start by observing that the angle that the (x_1, y_1)---(x_a, y_a) line makes with the line segment has to be a right angle. This means that the dot product between the vector $\langle x_a - x_1, y_a - y_1 \rangle$ and $\langle x_a - x_0, y_a - y_0 \rangle$ has to be zero. Equivalently,

$$(x_a - x_0)(x_1 - x_a) + (y_a - y_0)(y_1 - y_a) = 0 \qquad (5.6)$$

In addition, we know the relationship between x_0 and y_0, namely:

$$\frac{y_a - y_0}{x_a - x_0} = URM_1 \tag{5.7}$$

Substituting equation (5.7) into equation (5.6), we get the following:

$$(x_a - x_0) \cdot [(x_1 - x_a) + URM_1 \cdot (y_1 - URM_1 \cdot (x_a - x_0) + y_0)] = 0 \tag{5.8}$$

Solving equation (5.8) for x_a, we get the following closed form solution:

$$x_a = \frac{x_1 - URM_1 \cdot y_0 + URM_1 \cdot y_1 + URM_1^2 \cdot x_0}{1 + URM_1^2} \tag{5.9}$$

From equation (5.9), and equation (5.7), we immediately get the other half of the couplet:

$$y_a = URM_1 \cdot (x_a - x_0) + y_0 \tag{5.10}$$

One important thing to note is that since there are going to be more than 1 line segment in the plane, there will also be more than one possible (x_a, y_a) points that (x_1, y_1) can map to. Since we are only interested in mapping to the line segment that is nearest to (x_1, y_1), we only need to calculate the distances between (x_1, y_1) and all possible (x_a, y_a) pairs:

$$distance = \sqrt{(x_a - x_1)^2 + (y_a - y_1)^2} \tag{5.11}$$

and pick the appropriate pair.

Figure 5.7 below is an illustration of the distribution and movement of points as noise is added to the system, then denoised as explained in Section 3.6, then placed back onto the line segments as described by the projection method.

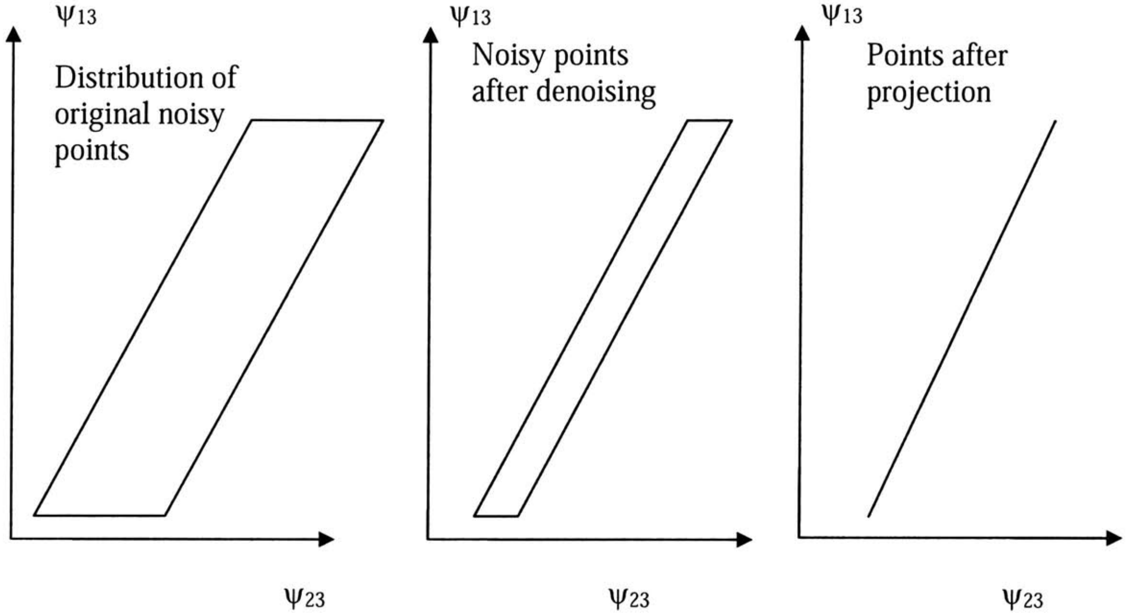

ψ13
Distribution of original noisy points
ψ23
ψ13
Noisy points after denoising
ψ23
ψ13
Points after projection
ψ23

Figure 5.7: Illustration of the distribution of points on the ψ_{13} - ψ_{23} plane before any processing, after denoising, and after denoising and projection.

Figure 5.8 is equivalent to Figure 5.3 with noise of 30 degrees, after we have performed the denoising step of Section 3.6, while Figure 5.9 shows the results after implementing projection.

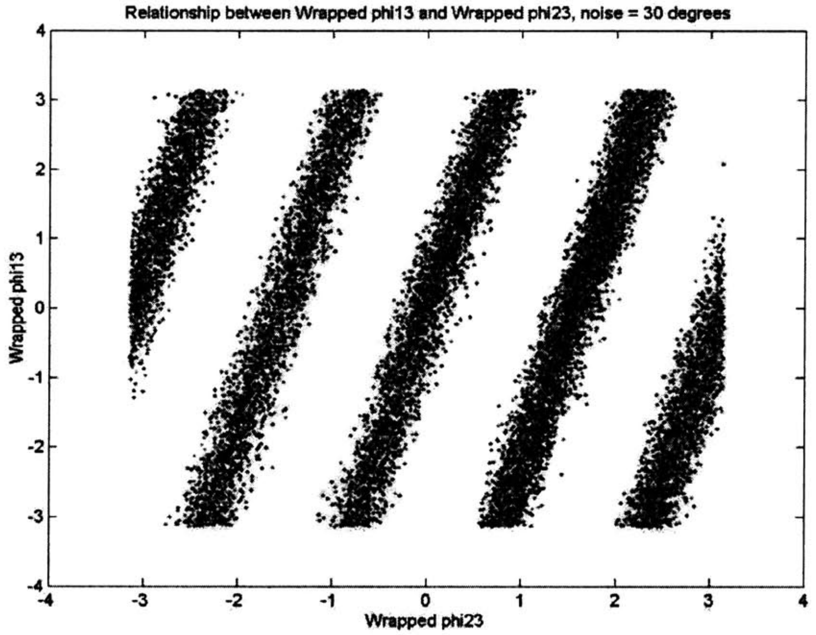


Figure 5.8: ψ_{13} and ψ_{23} after noise of 30 degrees is added.

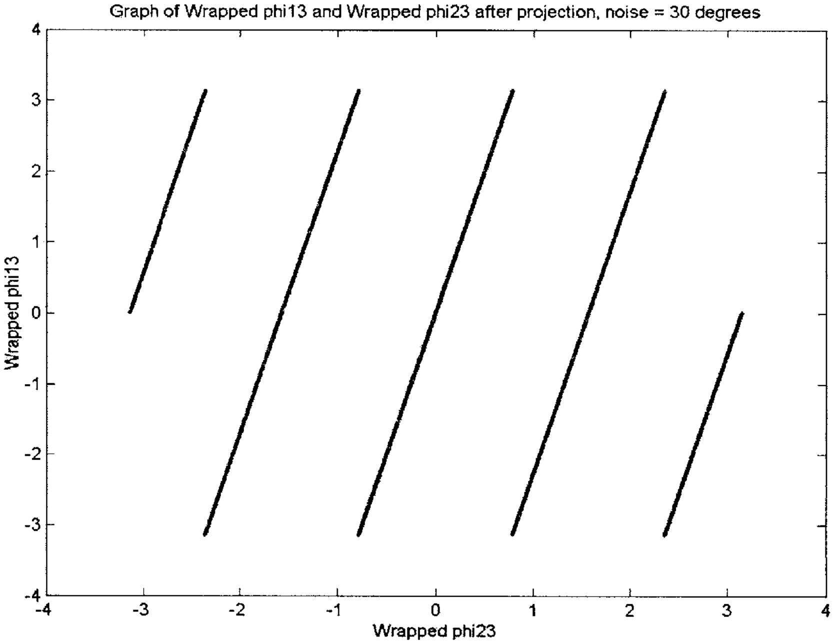


Figure 5.9: Figure 5.8 after points are projected onto line segments.

One question to be asked is: how do we even know the line segments in the first place? Since all we have are the noisy points, it is indeed difficult to pin down the exact location of the line segments. However, from Figure 5.8, we observe that we have so many points surrounding each line segment that taking the average over all those points should give us a pretty accurate estimate of where the line segments actually lie. Thus, we do not need to know the line segments in advance.

Another way that we can be sure of where the actual line segments is to observe the following: when we look at equations (5.5a) and (5.5b), we see that when the *original phase value* ϕ_{23} is zero, the *original phase values* ϕ_{12} and ϕ_{13} are both enforced to be zero so as to satisfy the constraints defined in the equations. This is an important observation; the point (0,0,0) in the ϕ_{23}-ϕ_{12}-ϕ_{13} space is the exact same point as (0,0,0) in the ψ_{23}-ψ_{12}-ψ_{13} space, since (0,0,0) does not undergo any wrapping at all. Using this information, we can set up our "reference point" at the origin, and start drawing the line segments from this one point. After those line segments are drawn, we can perform projection with ease.

From Figure 5.8, we can qualitatively see the inverse relationship between phase noise and effectiveness of Unambiguous Range Magnification. With a larger and larger URM value, the ψ_{13}-ψ_{23} plane gets more and more cluttered with line segments, which would mean a larger and larger unambiguous interval for the phases. In the noiseless case, this would be ideal. However, in the presence of noise, points are shifted away from the line segments. If URM is large, the high density of line segments implies that the distance between 2 adjacent line segments is small. If the noise is high or the URM is high, we could very well be moving points onto the wrong line segments since the segments are so close to one another. The assumption that the closest line segment is where the noisy point originally came from will therefore be wrong, and we would have introduced even more noise into the system than if we had just left the points where they were. Therefore, we cannot make the URM arbitrarily large without eating into our noise budget – a clear engineering trade-off.

5.5 What if the URM ratio is not an integer?

Up to this point, we have assumed that our baseline ratios (which are equivalent to the URM ratios by Equations (5.5a) and (5.5b) are integers. However, as mentioned before and also in [12], there will be a finite number of line segments as long as the URM ratio is in the set of rational numbers (of which the set of integers is a subset). We now investigate the scenarios under which URM is a non-integer rational number.

Since every rational number can be expressed as a ratio of 2 integers, we let URM=$\frac{p}{q}$, where p and q are both elements of the set of non-zero integers. Then, equation (5.5a) becomes the following:

$$\frac{\psi_{13}+k_2\cdot 2\pi}{\psi_{23}+k_3\cdot 2\pi}=\frac{p}{q} \qquad (5.12)$$

The k's merely denote that the phases are wrapped. After we link the line segments together and plot ψ_{13} against ψ_{23} (like in Figure 5.5), the resulting line will have a slope of $\frac{p}{q}$. This means that after linking, the unambiguous range for the horizontal axis is expanded $q\cdot 2\pi$ times, while that for the vertical axis is expanded $p\cdot 2\pi$ times. This has some obvious consequences:

1. If we want to expand the range of ψ_{13} by 4 times, we can use a baseline ratio of $\frac{4}{3}$ instead of $\frac{4}{1}$, which allows us to place the satellites much closer to each other and hence ensuring good coherence between the signals.

2. The drawback of using non-integer values of URM to achieve point 1 is that you now have a higher density of lines wrapped in the 2π by 2π box in the first place. From the previous section, we discussed that a higher density of lines leads to a degradation of noise budget.

In addition, if $\frac{p}{q}$ is a rational number close to an integer (e.g. 2.8), several line segments would appear in very close to one another. Figure 5.10 shows the line segments for B_{13} =

125m, B23 = 50m, which means a $\frac{p}{q}$ of $\frac{5}{2}$. Note the increased density of lines, as well as the increased unambiguous range for both the phase values of the horizontal and vertical axes. Figure 5.11 shows a $\frac{p}{q}$ of 2.8.

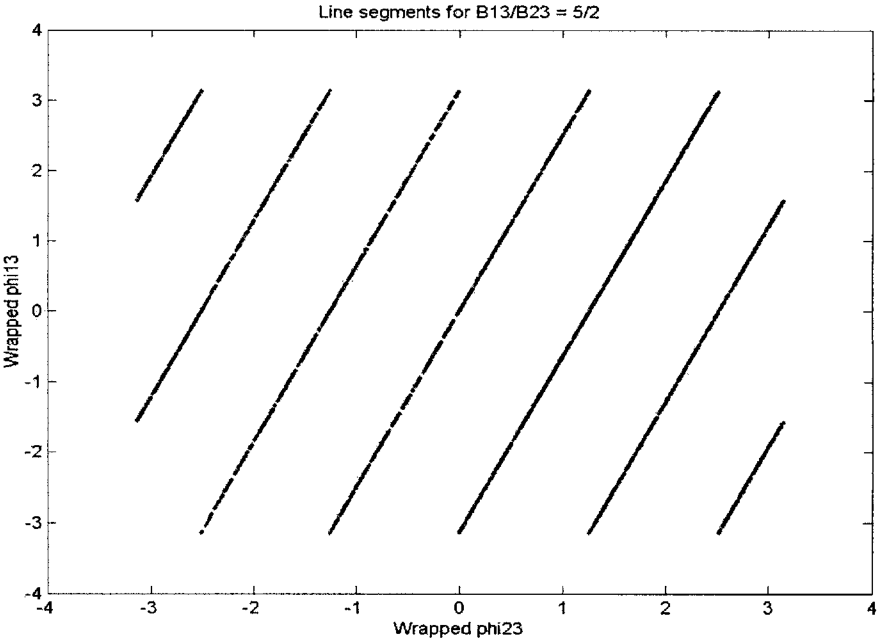


Figure 5.10: Line segments in the ψ_{13}-ψ_{23} plane with a URM = 2.5.

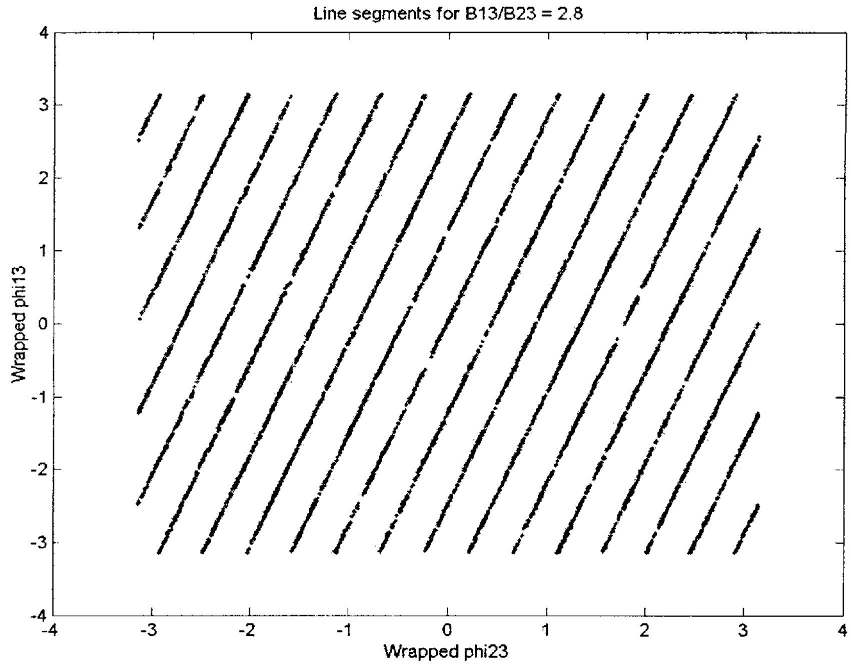


Figure 5.11: Line segments in the ψ_{13}-ψ_{23} plane with a URM = 2.8.

We introduce the notion of "noise distance" to describe the phenomenon of decreasing noise budget in exchange for a larger unambiguous range. The noise distance, d_n, is defined as the ½ of the shortest straight line distance (in radians) between 2 adjacent line segments in the 2π by 2π box. Qualitatively, this tells you how much the points can deviate from the line segments before they become too close to the adjacent line segment, and thus when adjusted, will get mapped onto a wrong line segment and thus a wrong ψ_{13}-ψ_{23} value pair.

The noise distance d_n can be derived as follows:

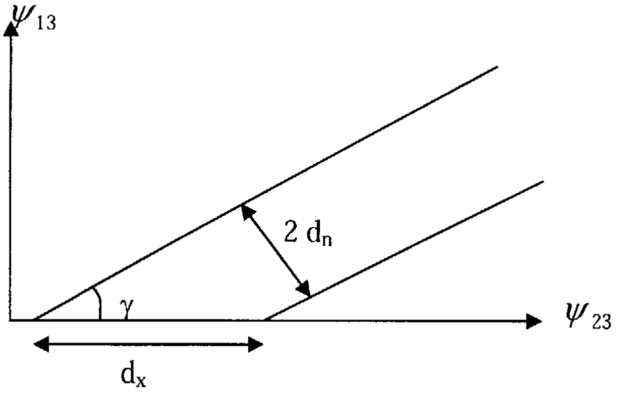


Figure 5.12: Illustration of the "noise distance" d_n.

We know that the line segments expand the vertical range by p times and the horizontal range by q times. By simple geometry and reasoning, it can be easily shown that after drawing all the line segments out, the horizontal distance between line segments d_x is given by:

$$d_x = \frac{1}{q}\frac{2\pi}{URM} \tag{5.13}$$

Also, the angle γ is given by:

$$\gamma = \tan^{-1} URM \tag{5.14}$$

Thus, by observing the geometry, we come up with the following expression for noise distance:

$$d_n = \frac{1}{q}\frac{\pi}{URM}\sin\left(\tan^{-1}(URM)\right) \tag{5.15}$$

The implication of equation (5.15) is clear. A small d_n (something we do NOT want) will appear if q is large, and that easily happens if our URM ratio is some rational number which has a large integer denominator when it is expressed as a ratio of integers. We plot a graph of noise distances against URM ratios below for URM ranging from 1 to 3.

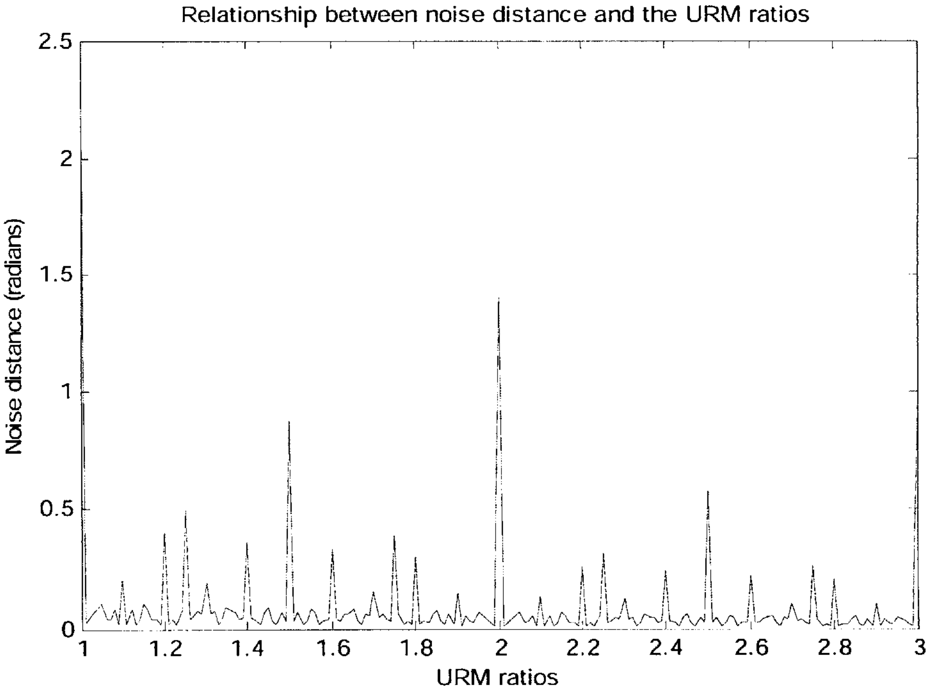


Figure 5.13: Plot of d_n against increasing URM ratios, illustrating the problem of non-integer URM values.

Figure 5.13 shows what we suspected – noise budget is really high for integer values and rational numbers that can be expressed as ratios of very small integers (e.g. 1, 2, 3, 1.5, 2.5. 1.2, etc), but it degrades significantly when your URM ratios become ratios of huge integer numbers.

As mentioned above, projection works best when baseline ratios are integer values, since only then can we clearly tell the correct mapping of noisy points back onto lines. For the purposes of exploring the usefulness of projection, we will first make the assumption that URM ratios are nice, round, small integers. This will obviously facilitate us checking to see if projection can live up to its promise in the most ideal of cases. Therefore, for the next few sections of this thesis, we use baseline values such that URM_1 and URM_2 are integers. However, after that discussion is complete, we will look into non-integer values of URM so as to get a more comprehensive view of 3-D Projection. All the results will be presented in this chapter of the thesis.

5.6 Projection method in 3 dimensions

Xu *et al* only suggested using a 2-D (i.e. ψ_{13} and ψ_{23}, ψ_{12} and ψ_{23}) plane to place points back onto the line segments. However, it is intuitive to see that we should not be projecting points in the ψ_{13}-ψ_{23} plane independently from those in the ψ_{12}-ψ_{23}; after all, they share the ψ_{23} axis, and you can easily get any phase difference value, e.g. ψ_{23} if you know the other 2 phase difference values, ψ_{13} and ψ_{12}. Projecting the points in the 2 planes independently will hence cause inconsistent phases that do not obey the geometry of the satellite equation. Although we are not using any value of ψ_{23} to construct our final heights, it seems that resolving this inconsistency may help to achieve more accurate results, since that further adheres to the geometry of the satellite configuration. As such, we propose to apply the projection method in the 3D space of ψ_{13}-ψ_{12}-ψ_{23}.

Consider the same ratio of baselines as before, but now we plot all the wrapped phases in the same 3D-space. The figures below highlight the inconsistency. 'o' is the original phase point without noise, 'x' is the noisy phase point, '◊' is the noisy point 2D-projected in the ψ_{12}-ψ_{23} plane, '∇' is the noisy point 2D-projected in the ψ_{13}-ψ_{23} plane, and 'Δ' is the noisy point projected in the 3D space.

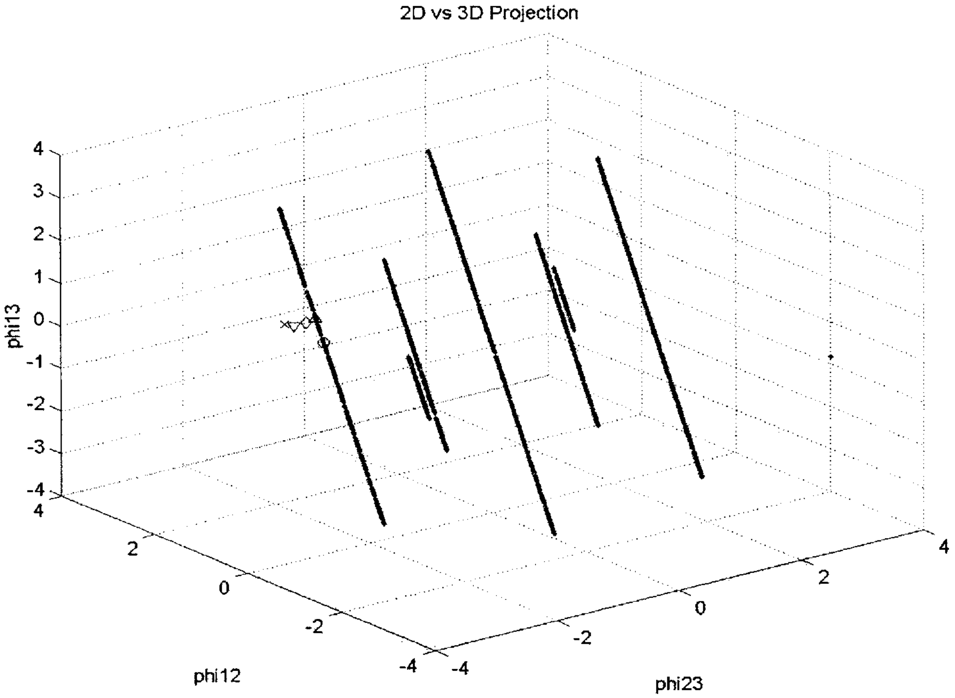


Figure 5.14: Noisy points before and after projection.

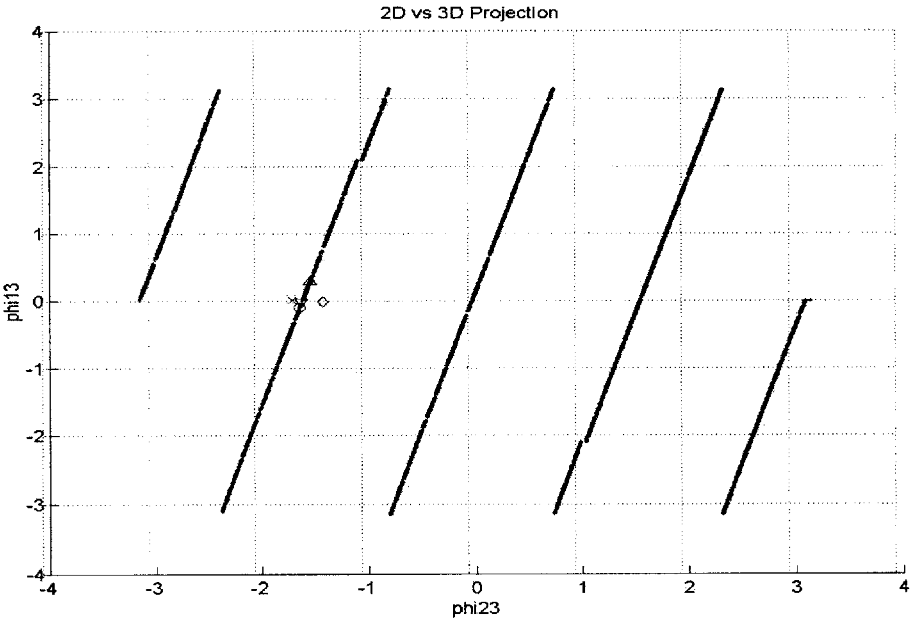


Figure 5.15: Looking at Figure 5.14 down the ψ_{13} - ψ_{23} plane.

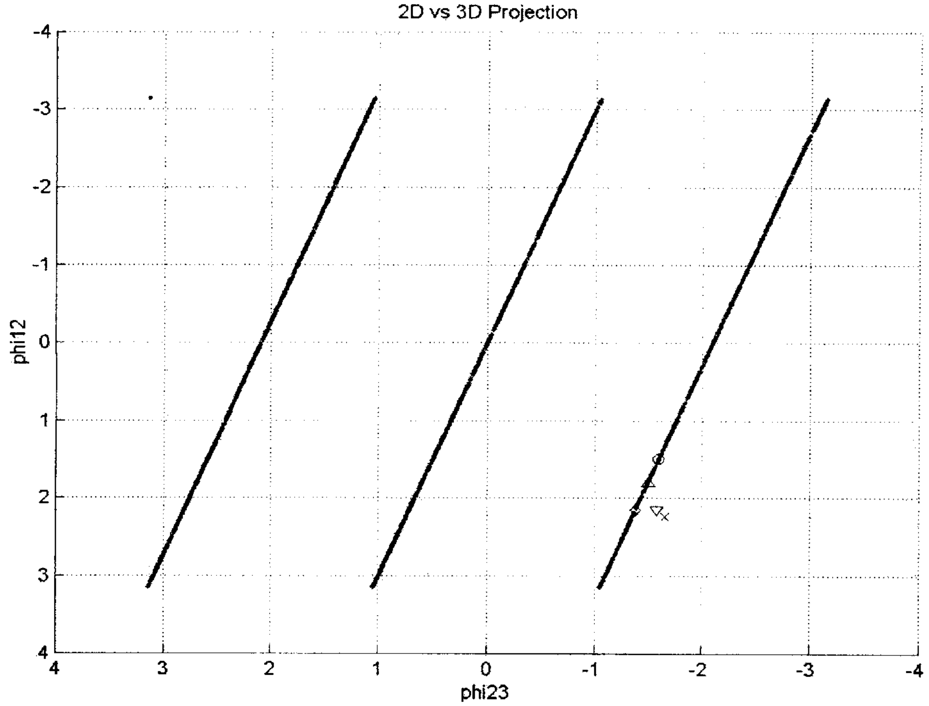


Figure 5.16: Looking at Figure 5.14 down the ψ_{12} - ψ_{23} plane.

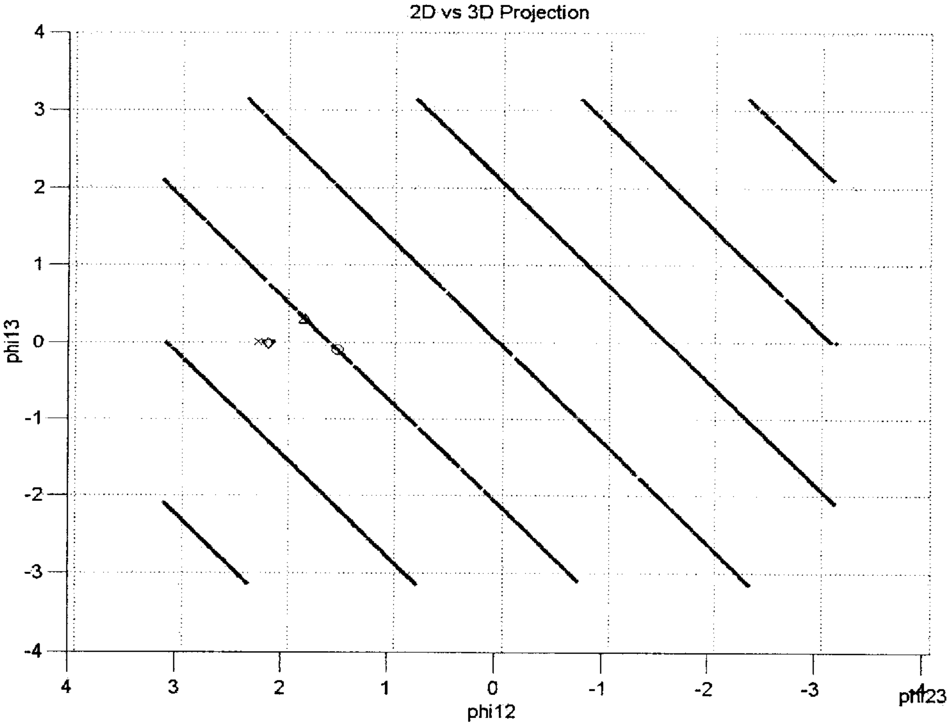


Figure 5.17: Looking at Figure 5.14 down the ψ_{13} - ψ_{12} plane.

It should be noted that if we were to rotate the point of view such that we are looking straight at the ψ_{13}-ψ_{23} plane (Figure 5.15), we would see the exact same thing that we would in Figure 5.3; the same thing goes for the ψ_{12}-ψ_{23} plane (Figure 5.16) and Figure 5.4. From Figures 5.14 – 5.17, we make the following observations:

1. 2D Projection moves the noisy points back onto the nearest line segment in the 2D plane.

2. When you view from another angle, the noisy point is really not on the line segment at all; it only LOOKS as if it is on the segment if you are looking down the correct plane.

3. 3D Projection truly moves the noisy point onto the line segment (the point is on the line no matter which angle you are looking at).

The line segments guarantee consistency as is defined by the satellite configuration (Equations 5.3a, b, c), and so only 3D projection satisfy the consistency requirement.

For ease of comparison, the line segments are shown again in Figure 5.18, while Figure 5.19 shows the noisy points scattered around the line segments. Figure 5.20 shows the noisy points after 3D projection is performed.

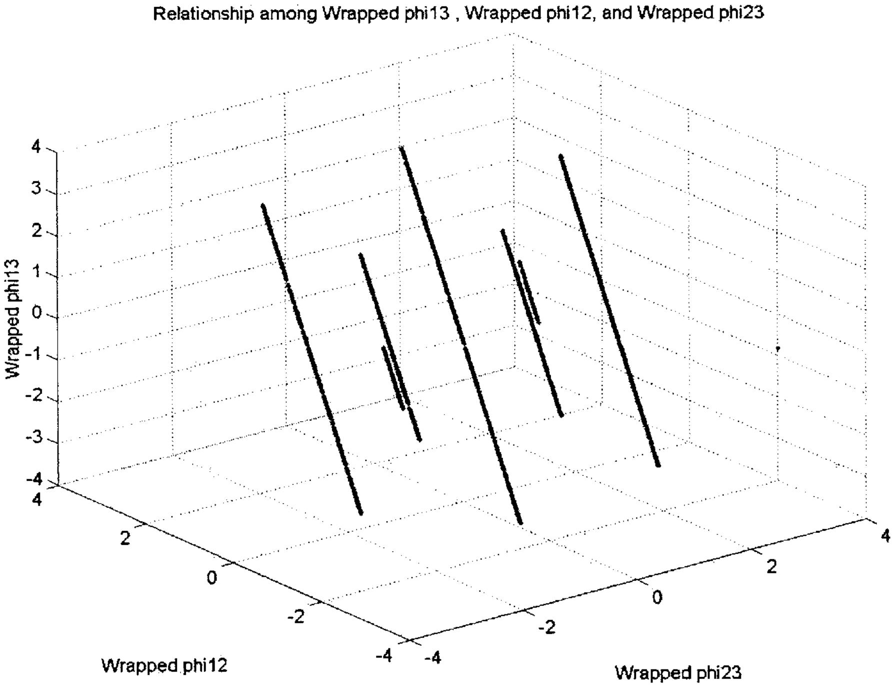


Figure 5.18: Line segments in the ψ_{13}-ψ_{12}-ψ_{23} space.

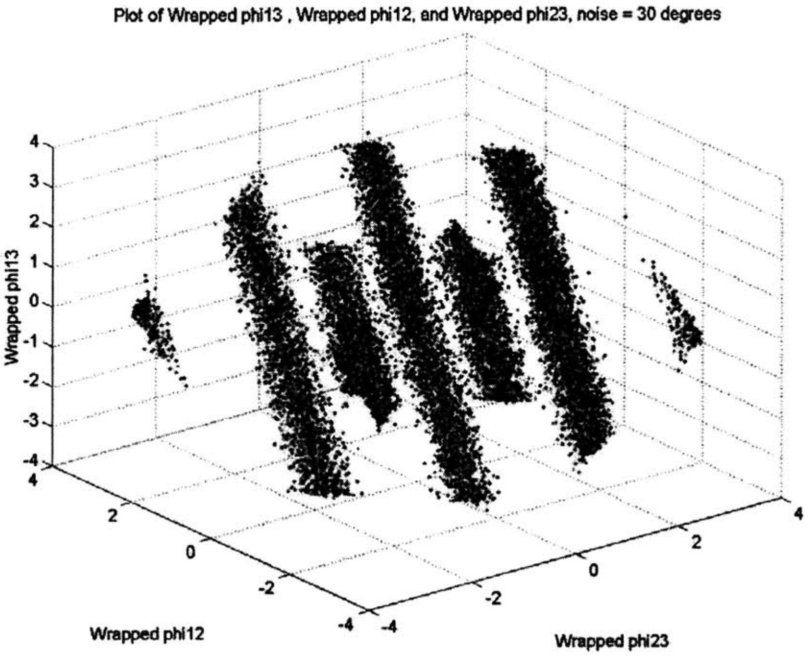


Figure 5.19: Noisy points in the ψ_{13}-ψ_{12}-ψ_{23} space (noise = 30 degrees).

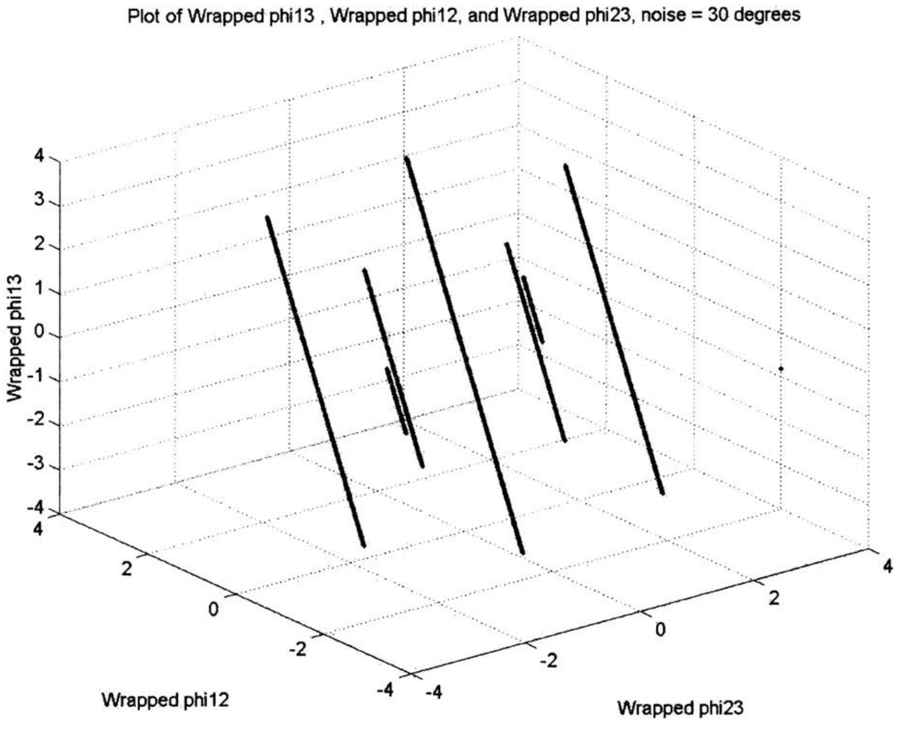


Figure 5.20: Figure 13 after points are projected back onto line segments.

Once again, just like 2-D Projection, we would want to frame 3-D projection in a solid, mathematical framework. Just like we did for 2-D Projection, we first define the following points that are of particular interest in this problem::

1. a known point on the line segment is defined as (x_0, y_0, z_0)

2. an actual ψ_{23}-ψ_{12}-ψ_{13} phase triplet defined by the interferogram is defined as (x_1, y_1, z_1)

3. The point that we want to find, i.e. the point on the line segment that is of the shortest Euclidean distance to (x_1, y_1, z_1) is defined as (x_a, y_a, z_a).

With that, we start by observing that the angle that the (x_1, y_1, z_1) ---(x_a, y_a, z_a) line makes with the line segment has to be a right angle. This means that the dot product between the vector $\langle x_a - x_1, y_a - y_1, z_a - z_1 \rangle$ and $\langle x_a - x_0, y_a - y_0, z_a - z_0 \rangle$ has to be zero. Equivalently,

$$(x_a - x_0)(x_1 - x_a) + (y_a - y_0)(y_1 - y_a) + (z_a - z_0)(z_1 - z_a) = 0 \qquad (5.16)$$

In addition, we know the relationship among x_0, y_0, and z_0, namely:

$$\frac{y_a - y_0}{x_a - x_0} = URM_1 \qquad (5.17a)$$

$$\frac{z_a - z_0}{x_a - x_0} = URM_2 \qquad (5.17b)$$

Substituting equation (5.17) into equation (5.16), we get the following:

$$(x_a - x_0) \cdot \begin{bmatrix} (x_1 - x_a) + URM_1 \cdot (y_1 - (URM_1 \cdot (x_a - x_0) + y_0)) \\ + URM_2 \cdot (z_1 - (URM_2 \cdot (x_a - x_0) + z_0)) \end{bmatrix} = 0 \qquad (5.18)$$

Solving equation (5.18) for x_a, we get the following closed form solution:

$$x_a = \frac{x_1 - URM_1 \cdot y_0 + URM_1 \cdot y_1 + URM_1^2 \cdot x_0 + URM_2^2 \cdot x_0 - URM_2 \cdot z_0 + URM_2 \cdot z_1}{1 + URM_1^2 + URM_2^2} \qquad (5.19)$$

From equation (5.19), and equation (5.17), we immediately get the other parts of the triplet:

$$y_a = URM_1 \cdot (x_a - x_0) + y_0 \tag{5.20a}$$

$$z_a = URM_2 \cdot (x_a - x_0) + z_0 \tag{5.20b}$$

One important thing to note is that since there are going to be more than 1 line segment in the plane, there will also be more than one possible (x_a, y_a, z_a) points that (x_1, y_1, z_1) can map to. Since we are only interested in mapping to the line segment that is nearest to (x_1, y_1, z_1), we only need to calculate the distances between (x_1, y_1, z_1) and all possible (x_a, y_a, z_a) triplets:

$$dis\,ta\,nce = \sqrt{(x_a - x_1)^2 + (y_a - y_1)^2 + (z_a - z_1)^2} \tag{5.21}$$

and then pick the appropriate triplet. All this is analogous to our mathematical formulation of 2-D Projection.

5.7 The boundary condition of the line segments

Projection seems to be a powerful idea for noise rejection. By making use of the satellite configuration as a guide, we were able to relate the different phase data from each individual satellite pair to one another so as to reject noise in the system. However, there is one subtle, though nevertheless critical, point that we have not taken into consideration. Observe the following figure, which illustrates a possible scenario when noise is added to a system.

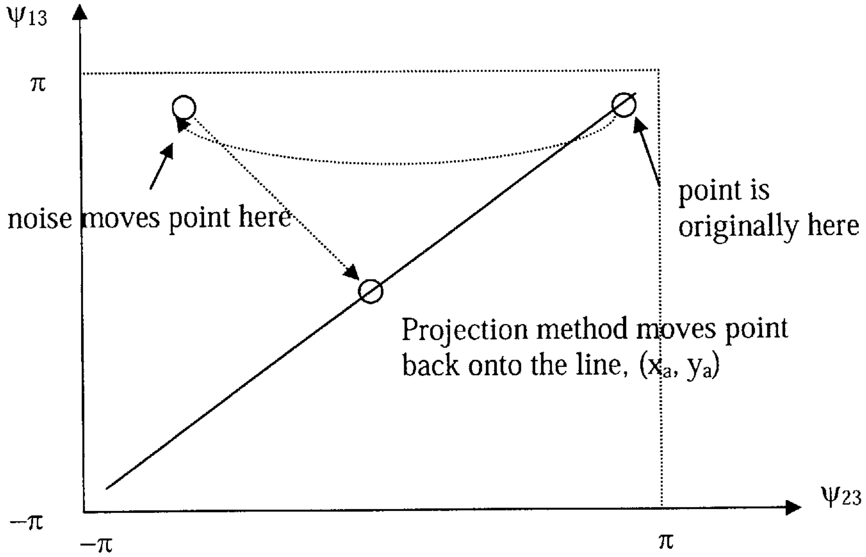


Figure 5.21: Illustration of a possible scenario during projection.

In Figure 5.21, imagine a point originally on the line segment as shown. A tiny bit of noise added to the system disrupts the point, such that the point is moved outside of the 2π by 2π box. Because of the wrapping process, the point is now mapped onto some place far away from where it should be. In essence, a tiny bit of noise has been magnified by the projection process.

This problem can actually be solved relatively easily, as long as we know that it does exist. In the scenario offered in Figure 5.21, suppose that we now introduce one more line segment **outside** of the 2π by 2π box. This new line segment is exactly like that shown in Figure 5.21, except that it's shifted to the left by 2π. Figure 5.22 below shows this.

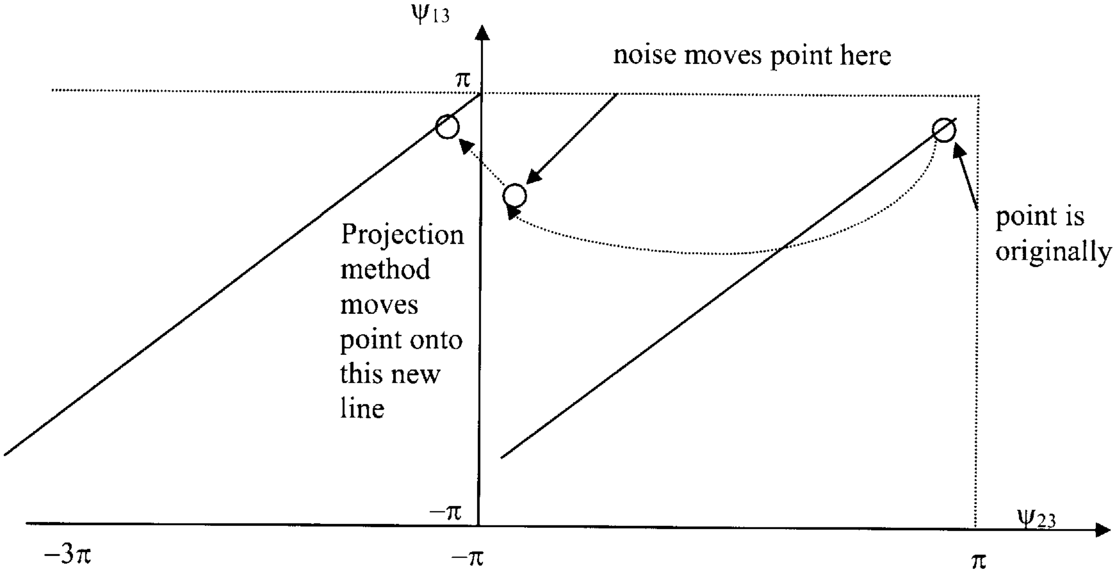


Figure 5.22: Illustration of Figure 5.21, with an extra line segment placed outside of original 2π by 2π box.

As shown in the figure above, the replication of the original 2π by 2π box in the region outside leads to a projection that, when wrapped back, produces results much closer to the original value. This insight is critical, as it further improves the value of projection in both the 2-D and 3-D algorithms.

We also have to recognize that the original 2π by 2π box has to be replicated in all directions so as to perform the best possible job in noise rejection. To accomplish such an end, this means that in 2-D Projection, we would need an extra 8 2π by 2π boxes in addition to the original one. In 3-D Projection, we would need an extra 26 2π by 2π by 2π cubes in addition to the original one. These additional boxes or cubes would have the same line segments (i.e. same slopes and same relative positions) as those in the original one, just shifted by the appropriate distances.

4.8 Is Projection alone just as good as Projection + URM?

In their paper [12], Xu et al suggested first projecting the points onto the line segments (in 2D phase plane), then using the expanded unambiguous phase range to get the actual phase values. If the terrain is not changing so rapidly such that it can be unambiguously represented within this expanded range, then we save ourselves the phase unwrapping step – clearly an advantage in terms of computational time and error induced from the unwrapping step itself. However, very often it is impractical to assume that we can do away with the actual unwrapping step, since we can never be sure if the terrain is changing that slowly, or whether our URM ratio is big enough. The question then is: If we do projection, link the line segments (expand range URM-fold), and unwrap in terms of URM times 2π, is that better than merely doing projection and unwrapping in terms of 2π? In essence, is the linking of line segments a redundant step?

The following figure shows a simulation of terrain, as well as the retrieved terrain height with both URM and Projection, and with only projection. The sub-picture in the figure is a blown-up view of the top portion of the "mountain".

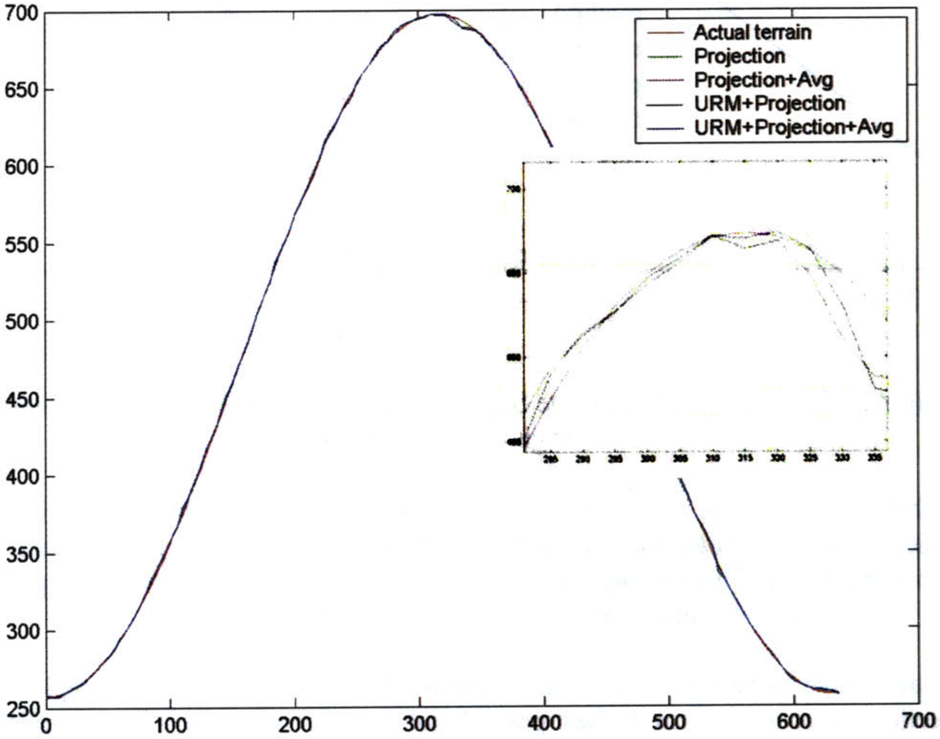


Figure 5.23: Comparing URM + Projection versus just Projection.

In Figure 5.23, we cannot see the green (Projection) and magenta (Projection + Averaging) lines because they are perfectly hidden behind the black (Projection + URM) and blue (Projection + URM + Averaging) lines. This goes for any "slice" of the 2D terrain we pick, and for any noise level we pick. We thus conclude that linking the line segments is a redundant step that does not improve our results when we unwrap using Weighted Least Squares.

5.9 Comparison of methods

Now, we are ready to compare the results of the following: just using WLS unwrapped phases from 1 interferogram (single baseline), arithmetic averaging across 2 longest baselines, WLS unwrapping with URM and projection in 3D space, and WLS unwrapping with URM and projection in 3D space before averaging. The results are presented in the table and plotted in the figure below.

Noise (deg)	SNR (dB)	Mean RMS Height Errors (m)					
		Single baseline	Average across 2 baselines	2D Projection	2D Projection with Averaging	3D Projection	3D Projection with averaging
10	15.16	0.4916	0.4832	0.4970	0.4859	0.4868	0.4907
20	9.14	0.6303	0.6038	0.6399	0.6096	0.6095	0.6174
30	5.62	0.8285	0.7737	0.8423	0.7620	0.781	0.7915
40	3.12	1.0639	0.9746	1.1053	0.9607	0.9743	0.9894
50	1.18	1.6327	1.3269	1.5768	1.2546	1.2645	1.2622
60	-0.40	2.7289	1.9173	2.5973	1.8307	1.6845	1.5982
70	-1.74	4.9736	3.2267	4.9566	3.1558	2.765	2.2935
80	-2.90	8.5898	5.7487	8.1582	5.4432	4.7123	3.8391
90	-3.92	14.1104	10.1253	14.035	9.9044	8.8414	7.2154

Table 5.2: Mean RMS height errors for different methods for 3 baselines

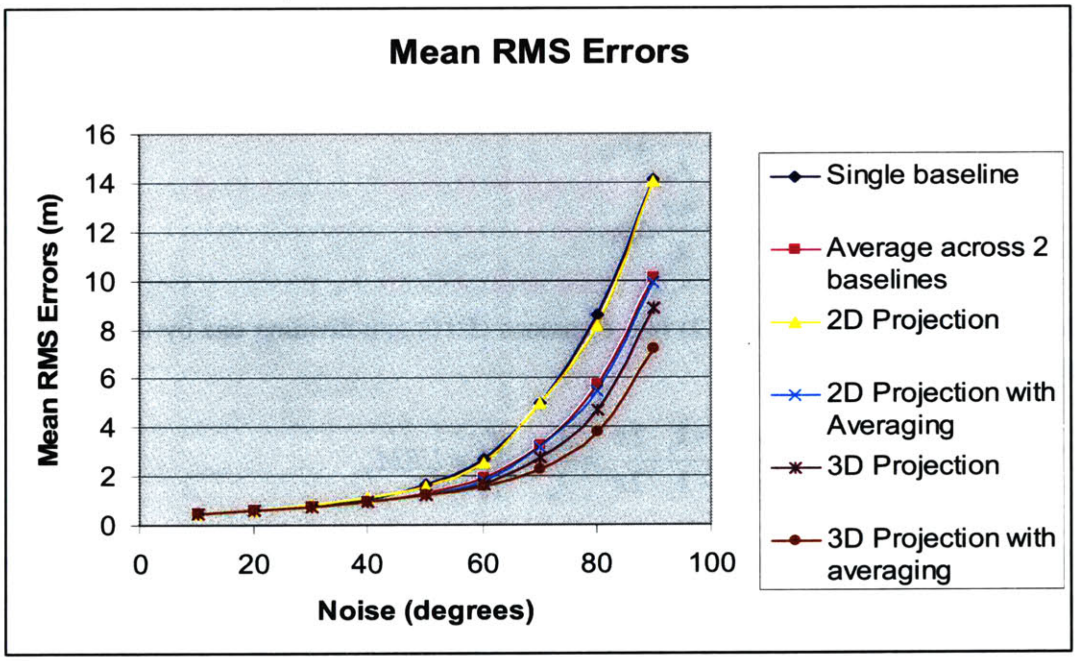


Figure 5.24: Plot of values in Table 5.2 as a function of noise.

From Table 5.2 and Figure 5.24, it is clear that in general, phase unwrapping after implementing unambiguous range magnification and the 3D Projection method performs better than simple averaging as suggested in [13]. In addition, URM and projection still stands to gain from averaging, which further increases the overall robustness of this process. For example, in the low SNR case of -1.74dB (70 degrees noise), the mean RMS error for 3D Projection with averaging is 83% that of 3D Projection, 73% that of 2D Projection with Averaging, 71% that of just simple averaging, and 46% that of just using one single baseline. For higher SNR cases, like the a SNR of 9.14 dB (20 degrees noise), all the algorithms do about just as good, which we would expect since the SNR is low and any method should work fine anyway.

In all simulations thus far, we only assume that our satellites make just 1 pass over the target terrain, i.e. we have managed to achieve such results with just one look. In contrast, Wong [14] showed that the maximum likelihood estimator (MLE) technique, first proposed by Lombardini [2], requires multiple looks to achieve results which are comparable with what the simple averaging technique can give in just 1 look. Both the MLE and the Projection method build on what the Unambiguous Range Magnification offers. It would thus seem that 3D Projection is able to give us much better results than other methods, including the maximum likelihood estimation.

5.10 Projection method with multiple frequencies

Equations (5.2a), (5.2b), and (5.2c) show that the phases are as related to frequencies, and hence wavelengths, as they are to baselines – phases are directly proportional to the frequencies in question. If we now use only 2 satellites (single baseline), but use 2 different frequencies – SAR$_1$ using f$_1$ and SAR$_2$ using f$_2$, we get the following sets of equations:

$$\phi_{12,a} = \frac{4\pi}{\lambda_a}\Delta\rho_{12} \tag{5.22a}$$

$$\phi_{12,b} = \frac{4\pi}{\lambda_b}\Delta\rho_{12} \tag{5.22b}$$

Now, substituting equation (3.2) into equations (5.10) and dividing one by the other, we get the following relationship:

$$\frac{\psi_{12,a} + k_1 \cdot 2\pi}{\psi_{12,b} + k_2 \cdot 2\pi} = \frac{\lambda_b}{\lambda_a} = \frac{f_a}{f_b} \equiv URM_f \tag{5.23}$$

Equation (5.23) looks almost identical to equations (5.5), except that now the 2 wrapped phases $\psi_{12,a}$ and $\psi_{12,b}$ are related by the ratios of frequencies used rather than baselines. Since the form of the equations is exactly the same as that of equations (5.5), we would expect all the reasoning behind the projection method to work just as well when we have 2 different frequencies and 1 baseline. In addition, it should be noted that equation (5.23) is exact; in contrast, equations (5.5) are approximate – they only hold if the property of similar triangles is observed, and that may not be true when the satellites get close enough to the terrain being illuminated.

Figure 5.25 shows the range magnification with $\frac{\lambda_b}{\lambda_a} = \frac{0.09m}{0.03m} = 3$. In reality, we usually don't have access to frequencies that differ by such a large amount, but this suffices as an example to show that URM holds for multi-frequencies as they did for multi-baselines.

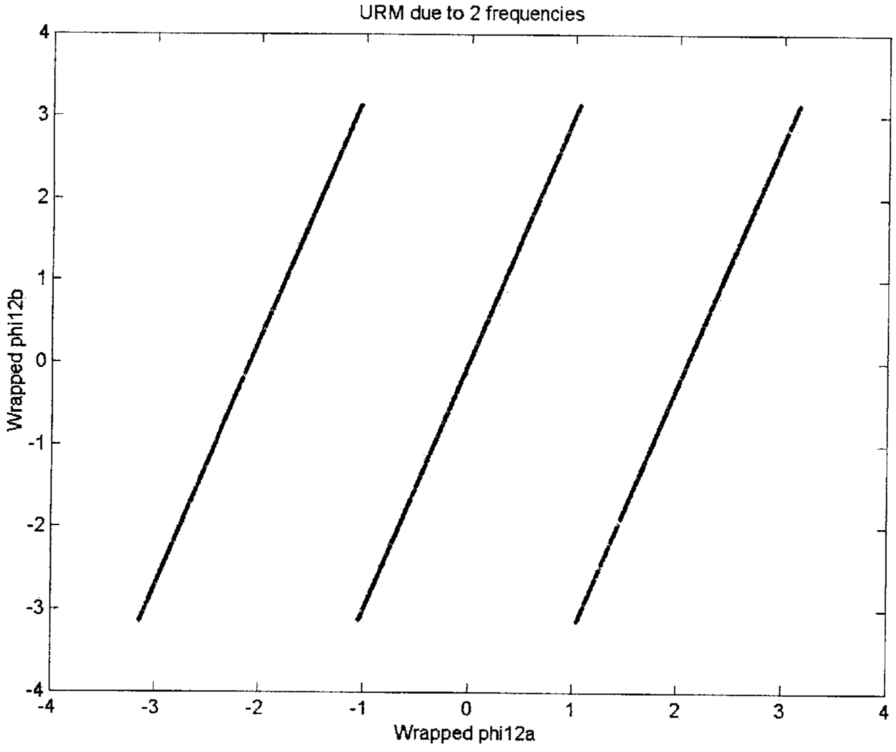


Figure 5.25: Relationship between $\psi_{12,a}$ and $\psi_{12,b}$.

Figure 5.25 looks exactly the same as Figure 5.4, which also has a URM of 3. Multiple frequencies thus promise to be a viable alternative to multiple baselines, since the equations turn out to be the same in the end. In addition, baseline lengths can never be made for certain since they have to be measured as satellites fly by one another; however, frequencies of signals originate from the transmitter, and that can be fine tuned to more precise values rather than taking a shot in the dark every time the terrain is illuminated.

Figure 5.26 shows noisy points being displaced from the line segments when there is 30 degrees of noise in the system.

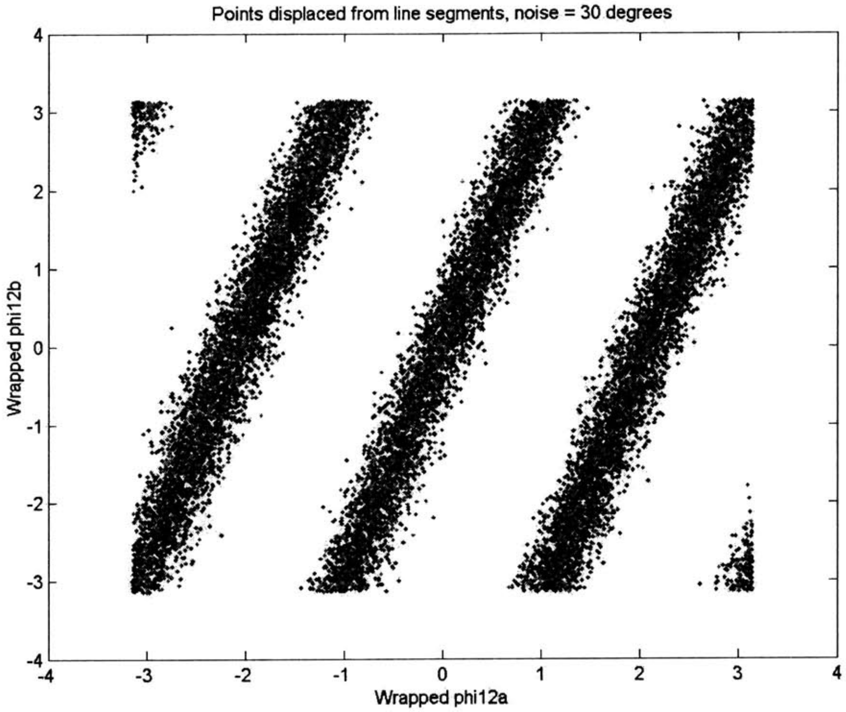


Figure 5.26: Phase points displaced by 30 degrees of noise, $\psi_{12,a}$ - $\psi_{12,b}$ plane.

Combining both multiple frequencies and multiple baselines, e.g. 3 satellites working with 2 frequencies into one framework should give us a few more degrees of freedom, since for that case every baseline can work at 2 different frequencies. More work has to be done to exploit that advantage, but it certainly is a promising prospect.

5.11 Special features of non-collinear satellite configurations

Up till now, we have assumed that the 3 satellites fit a collinear configuration, as shown in Figure 5.2. The systems of equations (equations 5.3 and 5.4) define this collinear geometry. However, it would be unreasonable to assume that the satellites fly in a perfectly collinear setup whenever they pass by some terrain. Therefore, we strive to formulate equations that capture the non-collinearity of satellites.

Figure 5.27 shows the non-collinear setup of the satellites.

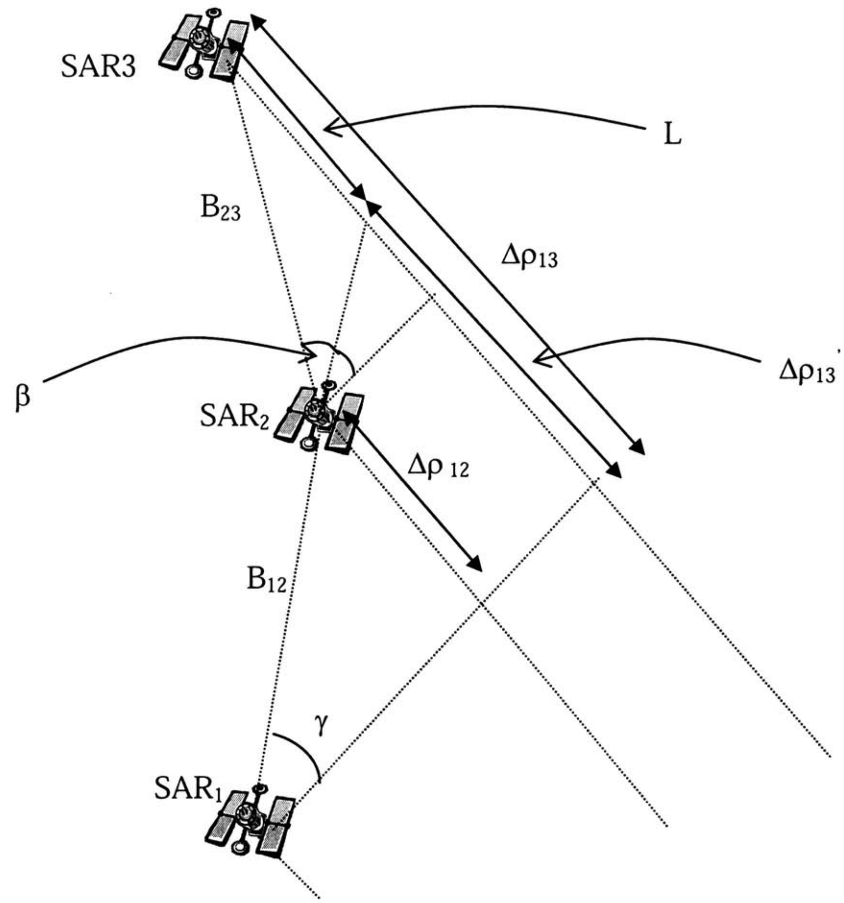


Figure 5.27: InSAR configuration for 3 satellites, non-collinear formation.

Like equations (5.3), the following hold true:

$$\phi_{12} = \frac{4\pi}{\lambda}\Delta\rho_{12} \tag{5.24a}$$

$$\phi_{13} = \frac{4\pi}{\lambda}\Delta\rho_{13} \tag{5.24b}$$

$$\phi_{23} = \frac{4\pi}{\lambda}\Delta\rho_{23} \tag{5.24c}$$

Using equation (3.2) and dividing (5.24b) by (5.24c), then making the appropriate substitutions based on the geometry of Figure 5.26, we get:

$$\frac{\psi_{13} + k_2 \cdot 2\pi}{\psi_{23} + k_3 \cdot 2\pi} = \frac{B_{12}\sin(\gamma)}{B_{23}\sin(\beta+\gamma)} + 1 \equiv URM_1 \tag{5.25a}$$

Equation (5.25a) is now in the same form as (5.5a), except that its right hand side is now a much more complicated expression instead of simple baseline ratios. Note that in the special case of collinearity, i.e. $\beta = 0$, (5.25a) reduces to (5.5a).

Performing the corresponding substitutions with equations (5.24a) and (5.24c) gives us:

$$\frac{\psi_{12} + k_1 \cdot 2\pi}{\psi_{23} + k_3 \cdot 2\pi} = \frac{B_{12}\sin(\gamma)}{B_{23}\sin(\beta + \gamma)} \equiv URM_2 \qquad (5.25b)$$

Once again, equation (5.25b) reduces to (5.5b) in the special case of collinearity.

Figure 5.28 below shows the line segments for a β angle of 15 degrees. This problem of non-collinearity manifests itself in the exact same manner as would non-integer URM's, since the URM's for equations (5.24a) and (5.24b) will rarely be integers even if the original baseline ratios are. Thus, solving the problem of non-collinearity will solve the problem of non-integer URM's, and vice versa, with a few subtle differences that we have to take note of. First, if we look at equations (5.25), sines are embedded in the equations. This naturally means that we can very easily get URM ratios that are irrational numbers, since the sine function regularly throws out irrational numbers (take $\sin\left(\frac{\pi}{4}\right) = \frac{\sqrt{2}}{2}$ as a good example). As we discussed before, projection, at least in its current state now, requires the URM ratio to be a rational number; otherwise, there would be an unlimited number of line segments that could threaten to fill up the entire 3-D space. Obviously, this presents a problem to the usefulness of 3-D Projection. We will go about trying to fix that problem later in this chapter.

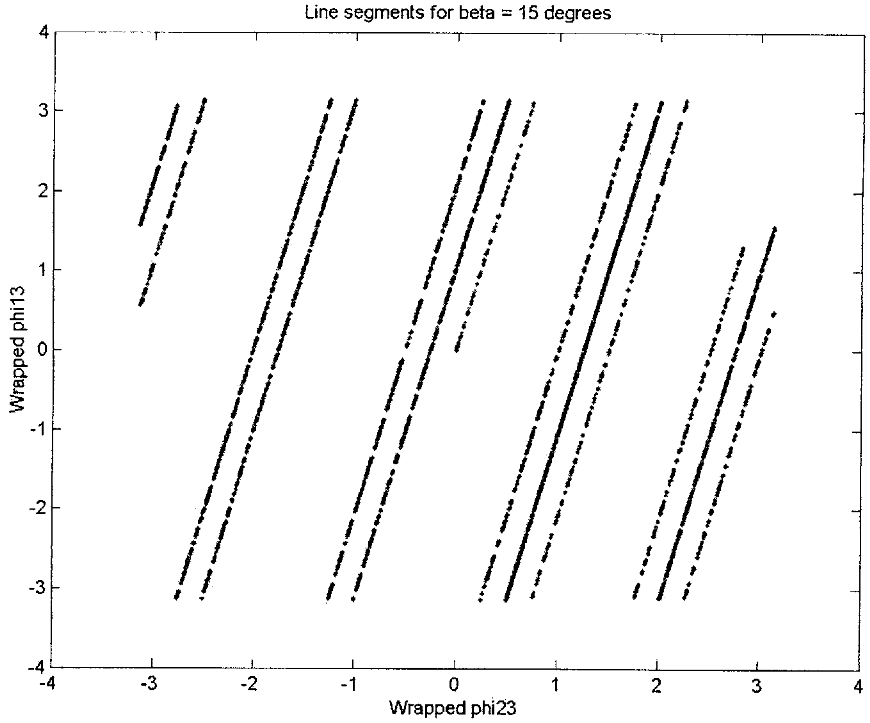


Figure 5.28: ψ_{13}-ψ_{23} plane for β=15 degrees.

Of perhaps graver concern, and a thoroughly more important difference between non-collinearity and collinear non-integer baseline ratios, is that the URM ratio in the non-collinear case might actually vary as a function of range distance. As each of the 3 (or more) satellites beam out to further and further distances in the range direction to map terrain that is further and further away from the original position of the satellites, something peculiar occurs as a direct consequence of non-collinearity → the URM ratio has to change. Observe the next 2 figures to see this in action.

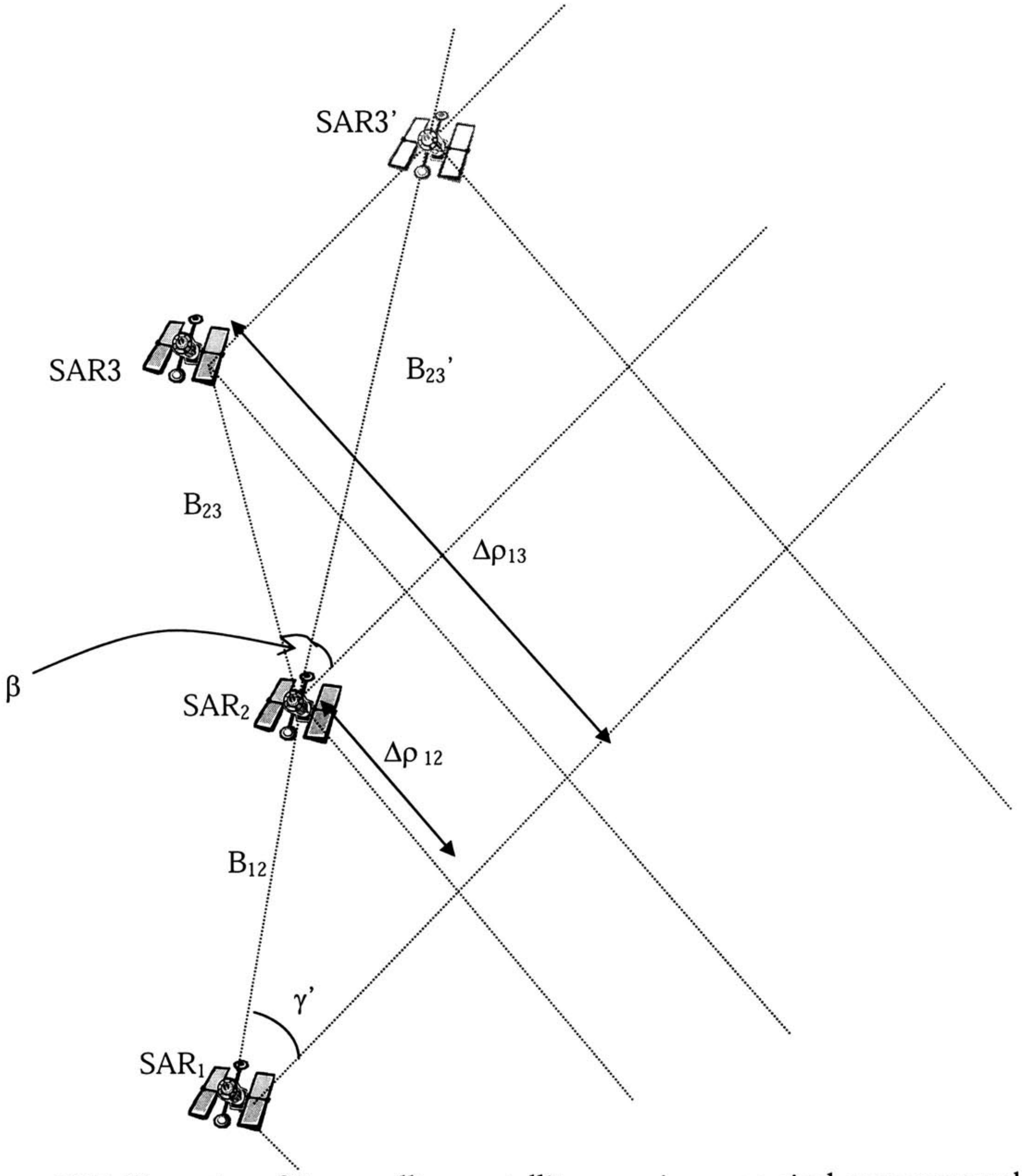


Figure 5.29: Illustration of 3 non-collinear satellites mapping some pixel on near ground.

In Figure 5.29, the satellites are mapping some pixel on the ground, and that particular pixel is at a range distance such that the angle that B_{12} makes with the arc of the circle that extend from the pixel is γ'. Suppose that there are 3 satellites flying in this system, namely SAR_1, SAR_2, and SAR_3. Since these 3 satellites are flying non-collinearly, we apply equations 5.25, and get the following:

$$\frac{\psi_{13} + k_2 \cdot 2\pi}{\psi_{23} + k_3 \cdot 2\pi} = \frac{B_{12}\sin(\gamma')}{B_{23}\sin(\beta + \gamma')} + 1 \equiv URM_1 \tag{5.26a}$$

$$\frac{\psi_{12} + k_1 \cdot 2\pi}{\psi_{23} + k_3 \cdot 2\pi} = \frac{B_{12}\sin(\gamma')}{B_{23}\sin(\beta + \gamma')} \equiv URM_2 \tag{5.26b}$$

However, we can also imagine the following scenario: instead of a SAR_3 flying non-collinearly with the other 2 satellites, there is a SAR_3' in the position shown in Figure 5.29 such that SAR_1, SAR_2, and SAR_3' form a collinear triplet. Since $\Delta\rho_{13}$ is identical for both the satellite configurations with SAR_3 and SAR_3', it has to be true that their interferograms are the same. Of course, this hinges on the assumption that the slant range vectors are essentially parallel to one another, so that the similar triangle assumption is approximately true. Taking that as a given, we can reach the following conclusion:

$$\frac{\psi_{13} + k_2 \cdot 2\pi}{\psi_{23} + k_3 \cdot 2\pi} = \frac{B_{13}}{B_{23}'} \equiv URM_1 \tag{5.27a}$$

$$\frac{\psi_{12} + k_1 \cdot 2\pi}{\psi_{23} + k_3 \cdot 2\pi} = \frac{B_{12}}{B_{23}'} \equiv URM_2 \tag{5.27b}$$

We have just defined the imaginary baseline between SAR_2 and SAR_3' as B_{23}'. In addition, we have expressed the URM ratios as a function of this imaginary baseline. Now, let's consider the case where the satellites now map a different, further pixel of the terrain, and investigate the consequence of that change on the URM ratios.

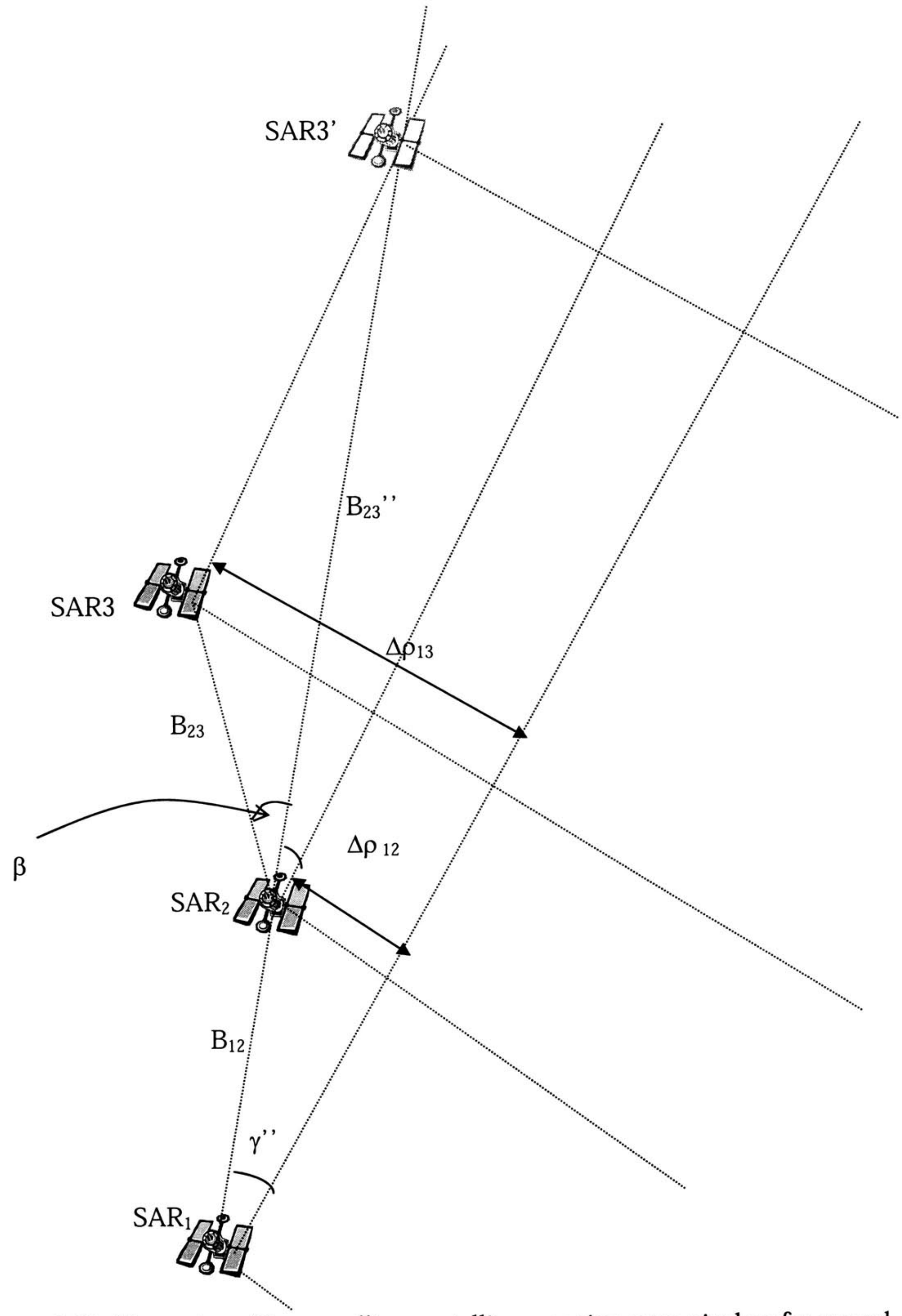


Figure 5.30: Illustration of 3 non-collinear satellites mapping some pixel on far ground.

Observing Figure 5.30, we can see that the slant range vectors now are tilted to a more horizontal position that they are in Figure 5.28. This is reasonable because the pixel being mapped is now further away from the satellites. Modeling after equations 5.26, we get the following:

$$\frac{\psi_{13}+k_2\cdot 2\pi}{\psi_{23}+k_3\cdot 2\pi}=\frac{B_{12}\sin(\gamma'')}{B_{23}\sin(\beta+\gamma'')}+1\equiv URM_1 \tag{5.28a}$$

$$\frac{\psi_{12}+k_1\cdot 2\pi}{\psi_{23}+k_3\cdot 2\pi}=\frac{B_{12}\sin(\gamma'')}{B_{23}\sin(\beta+\gamma'')}\equiv URM_2 \tag{5.28b}$$

Of course, we also observe that the imaginary satellite SAR$_3$'' is now much further from SAR$_1$ and SAR$_2$ than SAR$_3$' is in Figure 5.29. This naturally leads to the following formulation of URM ratios:

$$\frac{\psi_{13}+k_2\cdot 2\pi}{\psi_{23}+k_3\cdot 2\pi}=\frac{B_{13}}{B_{23}''}\equiv URM_1 \tag{5.29a}$$

$$\frac{\psi_{12}+k_1\cdot 2\pi}{\psi_{23}+k_3\cdot 2\pi}=\frac{B_{12}}{B_{23}''}\equiv URM_2 \tag{5.29b}$$

Looking at Figures 5.29 and 5.30 and equations (5.27) and (5.29), we make the following observations:

1. The satellites did not move in position as it scanned from a pixel in closer range to a pixel in further range.

2. However, since the imaginary baseline changed from B$_{23}$' to B$_{23}$'', the URM ratios changed as well, and this means that the line segments in the ψ_{13}-ψ_{12}-ψ_{23} space are no longer straight, but instead are curved lines since the slopes change as a function of phases.

3. Using the parameters in Table 3.1 and 5.1, we find that the change in slant look angle θ from one end of the terrain to the other end of the terrain varies by 0.05 degrees. Thus, for the test case terrain that we are using for our tests, we do expect the line segments to be approximately straight, and will carry out projection as if they are straight. However, it is critical to keep in mind that in cases of larger, wider terrain, this phenomenon manifests itself prominently, and has to be taken into account.

4. Collinear, non-integer baseline integer configurations do not suffer from this problem of changing URM's because the similar triangle property is always satisfied.

6. Projection for a Cartwheel Configuration of Satellites

6.1 Setting up the cartwheel configuration

Now, we step beyond what the previous report has considered and investigate the case where the satellites are not configured ideally --- the baseline ratios are not integers, and the satellites themselves do not line up in a collinear fashion. It is important to note that when we plot phase value ψ_{23} against ψ_{12} against ψ_{13} in a 3-D space, there will be straight lines of slopes determined by equations (5.19a) and (5.19b). In essence, for the purpose of this thesis, the problem of non-collinearity manifests itself in the exact same manner as would collinear non-integer baseline ratios. For this reason, in this section, we still only consider the case where the satellites are collinear but vary the baselines through different baseline ratios to obtain different values of URM's. This is just to keep the satellite simple, and is by no means limiting the value of 3-D Projection to merely collinear cases.

A large part of Wong's thesis [14] is devoted to exploring an optimization strategy using a satellite configuration whereby the satellites are fixed in a cartwheel, and rotate about a fixed point as they fly in the azimuthal direction. This cartwheel configuration has a unique advantage over other configurations → it allows us to gain access to otherwise seemingly impossible satellite combinations. For example, we are able to set $B_{13} = B_{12}$ in space, thereby overlapping satellites 2 and 3 in the azimuthal axis. This is shown in Figure 6.2 below. We utilize the cartwheel for a slightly different purpose → to obtain all the possible URM ratios. By rotating them as they fly, every single URM ratio can be obtained by virtue of the changing baseline ratios due to the rotation of the cartwheel. Figure 6.1 below shows this setup.

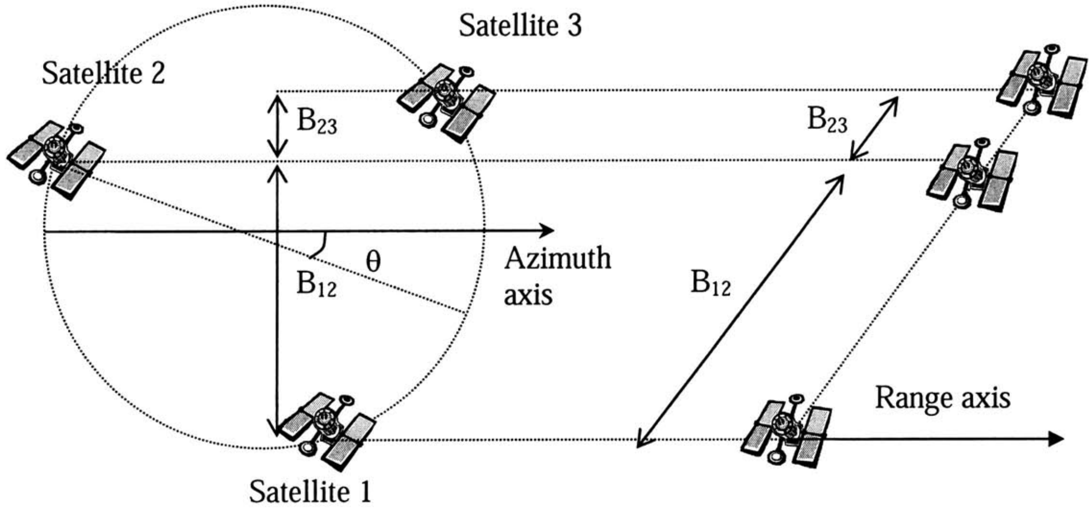


Figure 6.1: Illustration of cartwheel configuration that allows full range of URM values.

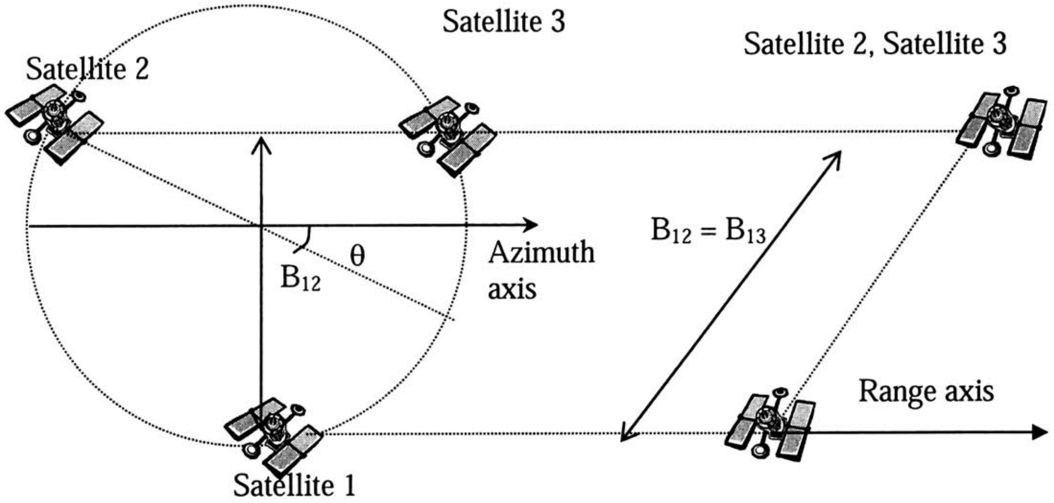


Figure 6.2: Illustration of special case, when satellites 2 and 3 overlap in the azimuthal axis.

One thing to take note of is that we have really assumed that the cartwheel is much, much smaller in size than any other distances that we care about, e.g. the range distance and the height of the satellites above ground. In such a scenario, the 3 satellites are essentially at the same point in space in the azimuth-height plane, and so the cartwheel configuration collapses to a two dimensional problem as desired. This approximation is good so long as that assumption holds true. However, if satellites start to get further and further away, serious problems may arise in our assumptions. For the remainder of the thesis, we will take it that those approximations hold.

First, we need a mathematical relationship that describes the transformation from the tilt angle θ to the URM ratios. Letting the shortest straight-line distance between any pair of satellites be R, it is fairly straightforward to show that the relationship is as follows:

$$B_{13} = R\cos(\theta) \tag{6.1}$$

$$B_{12} = R\cos\left(\frac{\pi}{3} - \theta\right) \tag{6.2}$$

$$URM_1 = \frac{B_{13}}{B_{23}} = \frac{2}{1 - \sqrt{3}\tan(\theta)} \tag{6.3}$$

$$URM_2 = \frac{B_{12}}{B_{23}} = \frac{1 + \sqrt{3}\tan(\theta)}{1 - \sqrt{3}\tan(\theta)} \tag{6.4}$$

From the geometry and the equations, it is clear that we can obtain all the URMs that we can possibly get just by sweeping θ from 0 to 30 degrees. All other tilt angles will also map to the same URM range, so we only consider θ ranging from 0 to 30 degrees. Plotting URMs against θ, as θ varies from 0 to 30 degrees, gives us the following plot:

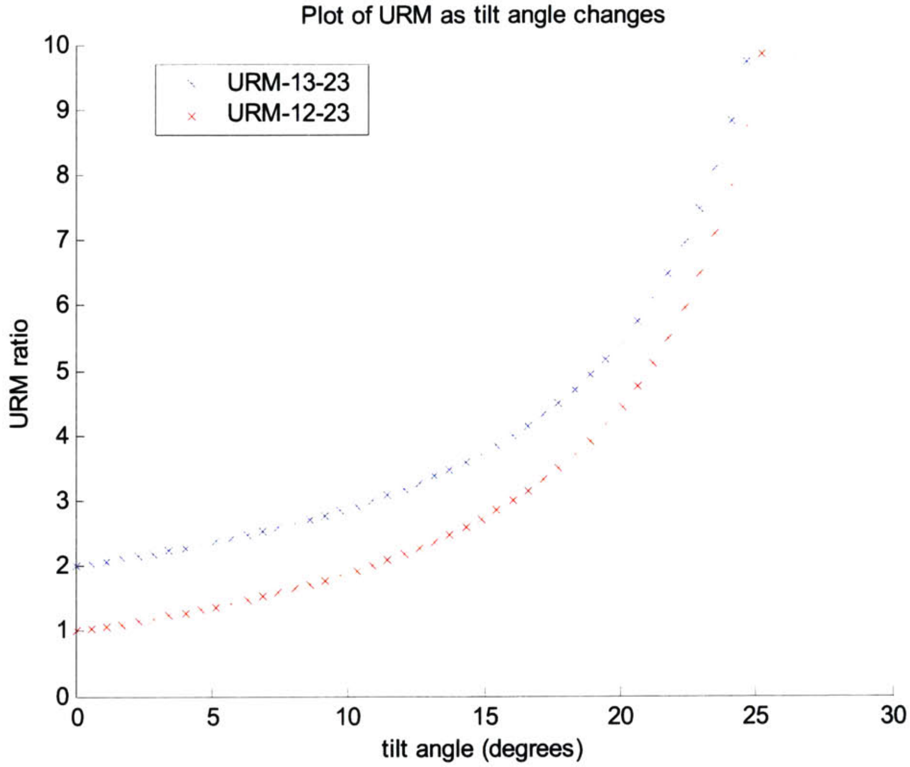


Figure 6.3: Plot of URM ratios as tilt angle changes.

We note that the URM ratios approach infinity asymptotically as we approach 30 degrees, because B_{23} gets increasingly small, and reaches 0 at $\theta = 30$ degrees.

6.2 Choosing an alternate URM ratio for projection

It is imperative to decide what URM ratios to actually use for any actual satellite geometry and configuration. For example, if the baselines are such that the ratio is 3.12648, do we use that actual ratio (and thus have lots of line segments), or do we choose an alternate URM ratio to do the projection? We seek to investigate this by performing the following experiment:

1. Change the tilt angle such that URM_1 is 3.35.

2. During the projection module, map the noisy points onto the line segments defined by URM_1 and URM_2, where $URM_1 \in \{3.1, 3.2, 3.3, 3.4, 3.5, 3.6\}$, and $URM_2 = URM_1 - 1$.

3. After mapping onto the chosen URM values, find the mean RMS error in height.

4. Repeat for all other URM values, until the set is exhausted.

We then plot the mean RMS errors in height as a function of URM$_1$, and present the results in the following figure.

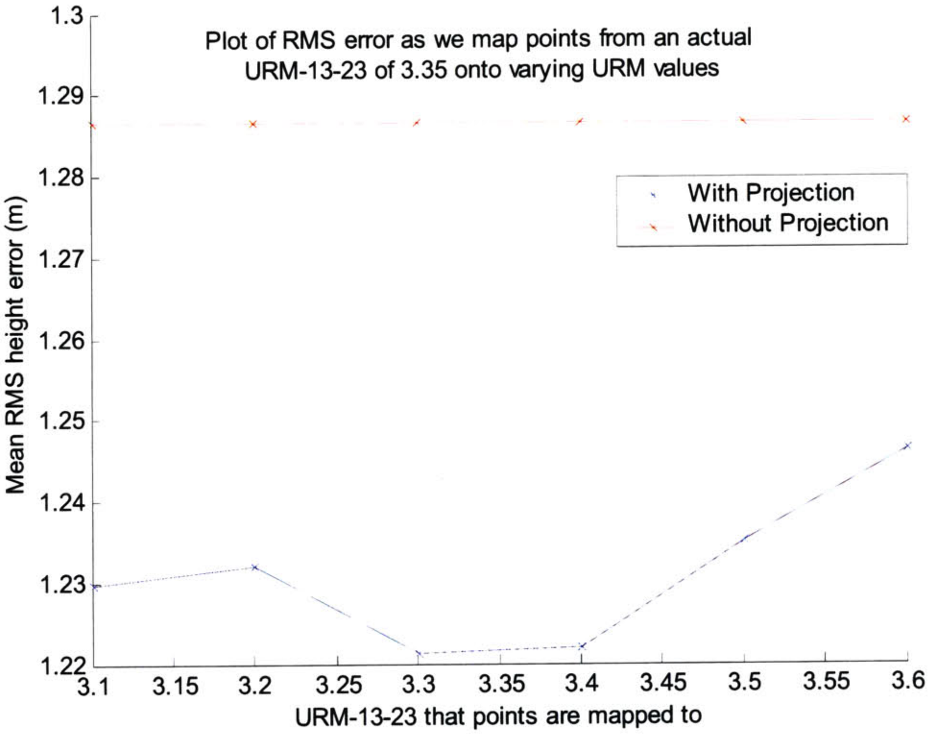


Figure 6.4: Illustration of RMS error as we map points onto different URM values.

From the figure above, we can see that we get the lowest RMS height errors when we map onto URM values closest to 3.35 (3.3, 3.4), and the results degrade as we map onto values further away from 3.35. This shows that we should always try to map onto URM values that are closest to the actual URM ratio. Intuitively, this makes sense, since the mapping process inherently disturbs the data and so might introduce noise of its own as well, and mapping data points onto a URM value that is significantly different from the actual URM based on the satellite geometry will inevitably introduce more noise.

So, now we have the algorithm for 3-D Projection as follows:

1. From the geometry of the satellite configuration, determine the actual URM values.

2. Determine the URM values to be used for projection. Mathematically, this is:

$$URM_{to-be-used} = \frac{\lfloor 10 \cdot URM_{actual} \rfloor}{10}$$

   This really just means rounding the actual URM value to the nearest 0.1.

3. Map the noisy points to line segments of the $URM_{to\text{-}be\text{-}used}$ values in 3-D space.

4. Continue with the height retrieval process using these newly corrected phase values.

6.3 Projection in the presence of noise

We have already seen the detrimental effects of noise on a system. Increasing levels of noise invariably introduces increasing errors in the heights retrieved from the phases of the interferogram. In the previous chapter, we have conclusively shown that with collinear satellites with integer baseline ratios, 3-D projection can filter out a large chunk of noise. Used in tandem with other noise filtering techniques like wavelet denoising or complex averaging, significant improvements can be expected. Now, we would like to investigate the merits of 3-D projection in a more general case → that of when URM ratio is not such a "nice" number anymore. Using a satellite configuration of URM_l = 3.35, we investigate the merits of 3-D Projection by finding the RMS height errors as we change the noise level. The results are shown below:

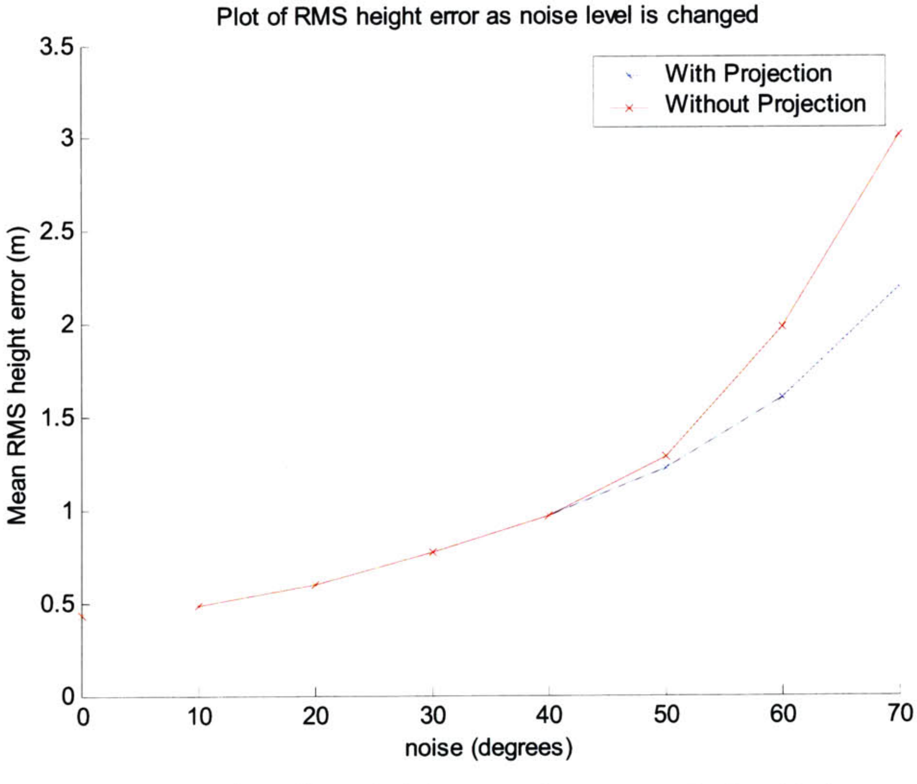


Figure 6.5: Illustration of how height errors change as noise level is increased.

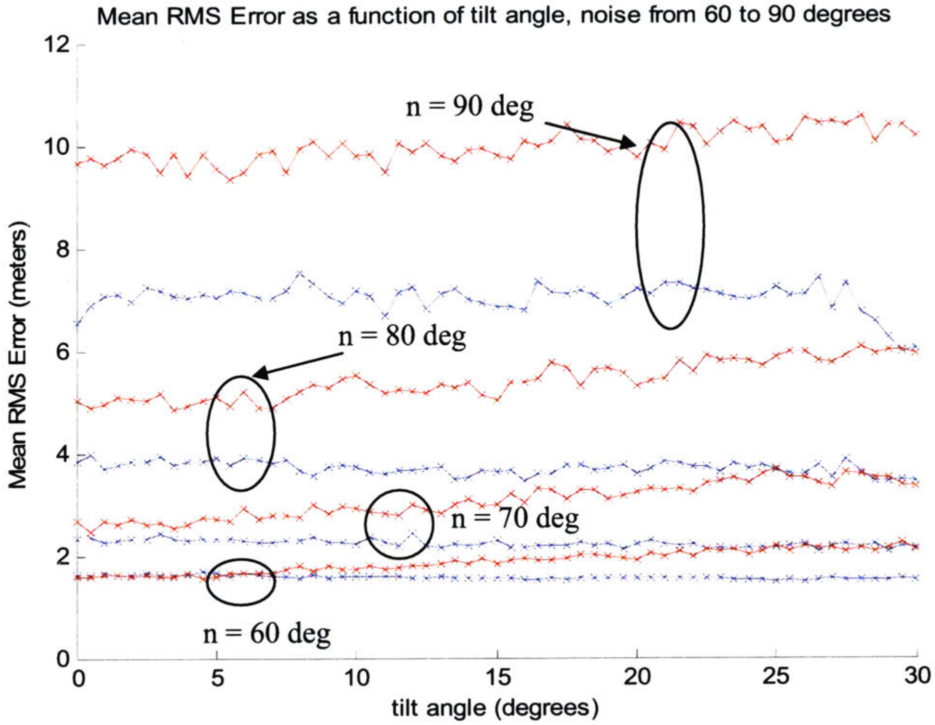
From the figure above, it is clear that 3-D projection does not fare worse than not-using Projection; in fact, for higher noise levels, the noise reduction capability of 3-D

Projection is fairly significant. In the cases of when noise level is fairly low (between 0 and 40 degrees), it is inconclusive from the graph whether projection helps or hurts noise filtering. However, the 2 graphs (red and blue) are both basically lying on top of each other in that noise range that even if using projection does hurt in that noise range, the effect is fairly insignificant. It is worth noting that even at a noise level of 0 degrees, there is still some height error. This is due to the fact that complex averaging is used as a denoising technique when the interferograms are retrieved. It has been shown that complex averaging does give much better results, but also inadvertently introduces errors when there's no noise.

6.4 Projection vs. no Projection, non-integer URM ratios

We have determined that we can rotate through from 0 to 30 degrees to get any combination that we want, and that projection in the presence of high noise level is still a viable, attractive option even with the URM ratios are not integers.

We now look at the relationships among the satellite configuration, i.e. as the cartwheel is rotated through different angles, the amount of noise in the system, and the mean RMS errors in height. Fifty random and different values of URM (and thus 50 different satellite configurations) are picked from within the set in the set $[0,30)$ degrees (which will provide us with any URM value in the set$[2,\infty)$). We vary noise values from 0 to 90 degrees to test for behavior of the different methods as noise increases. The results are shown in the following diagrams.



Figures 6.6: Illustration of how mean RMS height error changes as cartwheel is rotated and as noise in the system is changed. Blue lines are for cases when 3-D Projection is used, and red lines for cases when it's not used.

From the figure above, the following is clear:

1. At low noise levels, 3-D Projection even in the general case of non-integer URM ratios still does not introduce more error than if we do no projection at all.

2. At higher noise levels, 3-D Projection fares significantly better in terms of RMS height errors.

3. At tilt angles close to 30 degrees, the RMS error due to 3-D Projection seems to decrease significantly. This can be reasoned as follows: even though there is the rapidly shrinking B_{23}, which directly leads to a huge URM_{13-23} and URM_{12-23} and thus short noise distances, this effect does not show itself because of weighted averaging.

6.5 2-D Projection as a special case of 3-D Projection

- Up till now, we have been doing 3-D Projection as follows:

  - Given a point (x_1, y_1, z_1),

- Find the point x_a on the appropriate line segment that point (x_1, y_1, z_1) maps to.
- Find y_a, z_a from x_a by multiplying by appropriate URM ratio and adding the appropriate constants (equations 5.20).
- We have always been using ϕ_{23} as x. This is an arbitrary choice in the beginning, and is not normally expected to lead to any different behavior as when we use ϕ_{12} or ϕ_{13} as x. However, as we shall see, there are some subtle differences that may prove pivotal.
- For ease of comparison, we repeat equation (5.19) here, which sums up the closed form solution of x_a for 3-D Projection:

$$x_a = \frac{x_1 - URM_1 \cdot y_0 + URM_1 \cdot y_1 + URM_1^2 \cdot x_0 + URM_2^2 \cdot x_0 - URM_2 \cdot z_0 + URM_2 \cdot z_1}{1 + URM_1^2 + URM_2^2} \tag{6.5}$$

Now, we note that 2-D Projection is carried out in a very similar fashion, and in all our simulations except in Section 5.5, we have been using ϕ_{23} as x as well. This is once again an arbitrary choice made in the beginning. We repeat equation (5.9) here, which sums up the closed form solution of x_a for 2-D Projection:

$$x_a = \frac{x_1 - URM_1 \cdot y_0 + URM_1 \cdot y_1 + URM_1^2 \cdot x_0}{1 + URM_1^2} \tag{6.6}$$

By comparing equations (6.5) and (6.6), it should be clear that if we let all URM$_2$'s in (6.5) tend to zero, we end up with (6.6). This is mathematical observation, irregardless of what the variables in the equations stand for. Thus, mathematically, **with the variables meaning the same thing in both equations**, 2-D Projection is a special case of 3-D Projection.

However, so far, the symbols have been meaning different things in Section 6.5, when we used 2-D and 3-D Projection in tandem. In that simulation, x represents ϕ_{23} in 3-D Projection, but x represents ϕ_{12} in 3-D Projection. This means that equation (6.5) will not decompose into (6.6), because all the variables represent different things. However, if we make an adjustment and change the variables in the following manner for 3-D Projection:

- ϕ_{12} as x_a.
- ϕ_{13} as y_a.
- ϕ_{23} as z_a.
- URM$_1$ becomes the $\frac{B_{13}}{B_{12}}$ ratio.
- URM$_2$ becomes the $\frac{B_{23}}{B_{12}}$ ratio.

Now, as tilt angle approaches 30 degrees, B$_{23}$ approaches zero, and hence URM$_2$ approaches zero. Thus, the problem smoothly decomposes into a 2-D Projection picture from a 3-D Projection one. We can actually prove this experimentally with a simulation. Let ϕ_{12} be x_a, and ϕ_{13} be y_a. 2-D Projection on the ϕ_{12} - ϕ_{13} plane is used, and the results are juxtaposed on top of the results shown in Figure 6.6 b. The figure below shows this trend.

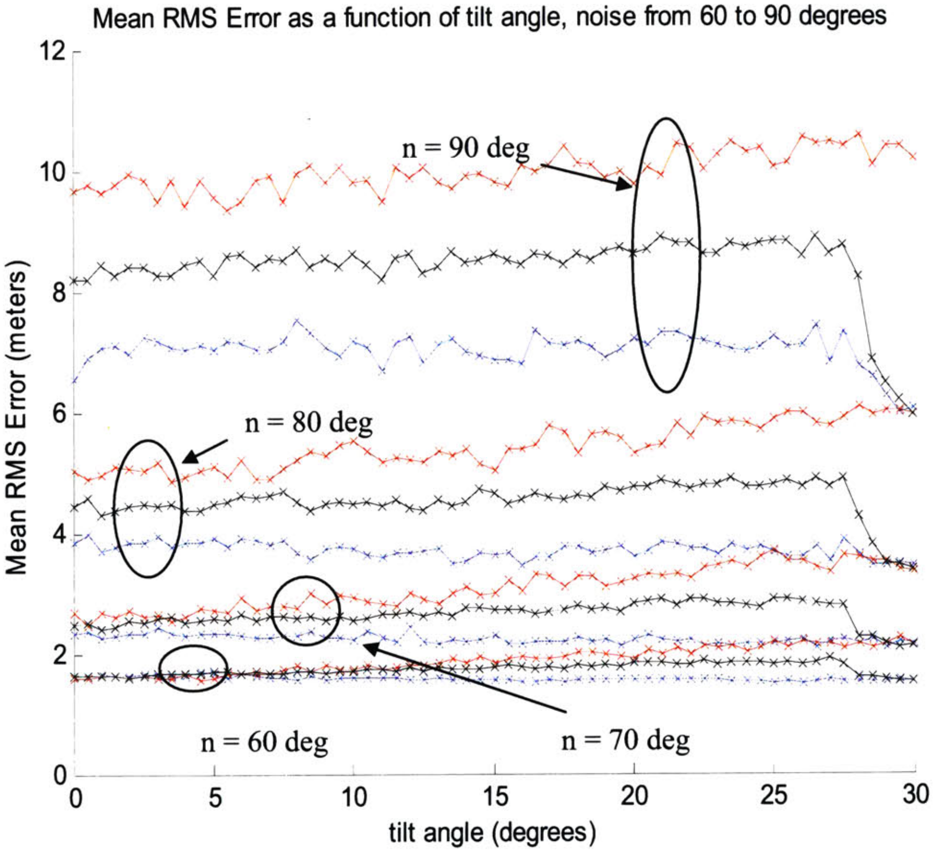


Figure 6.7: Illustration of how mean RMS height error changes as cartwheel is rotated and as noise in the system is changed. Blue lines are for cases when 3-D Projection (using ϕ_{23} as x) is used, black lines for cases when 2-D Projection is used in the ψ_{13} - ψ_{12} plane, and red lines for cases when no projection is used.

From Figure 6.7, it is clear that 2-D Projection follows the trend of rapidly decreasing RMS noise error as we approach a tilt angle of 30 degrees, which equates to SAR_2 and SAR_3 being one on top of the other. This simulation essentially proves our case that mathematically, 2-D Projection is but a special case of 3-D Projection. More importantly, the figure shows that 3-D Projection is a powerful noise-reduction tool for many cases. For example, comparing the cases of no projection used with 3-D Projection used for a noise level of 90 degrees, we see that for most tilt angles 3-D Projection is able to reduce RMS height errors by close to 30 percent, from 10 meters to 7 meters. That is a huge reduction in error, and one that makes 3-D Projection a very promising algorithm.

7. Summary

7.1 Summary of results

Throughout the course of this report, we have investigated the following problems:

Classical Phase Unwrapping Methods

- With only 2 antennas in the SAR configuration, we established the framework of the Unweighted Least Square algorithm, Weighted Least Square algorithm, Branch Cut algorithm with Cleaning, and Weighted Least Square with Cleaning. For the least square algorithms, presence or absence of residues is used as the main criteria for assigning weights.

- Weight assignment is based on a binary system, i.e. either 1's or 0's, and not on a linear scale.

- Weighted Least Square with Cleaning does much better than the other 3 other algorithms in phase unwrapping for our purposes.

Dynamic Programming with the MAP approach

- Dynamic Programming with the MAP approach can be done for both single baseline and multiple baselines. For the multiple baseline scenario, two underlying assumptions - constraints on elevation angle α and geometric ratios of satellite configurations - have to be made for the algorithm to function correctly as desired.

- Weighted Least Square with averaging still does better than multibaseline MAP or single baseline MAP with averaging. It thus would lead us to conclude that Weighted Least Square unwrapping with denoising, 3D Projection, and averaging would be the best algorithm in going from interferogram to heights.

Fundamentals of 2-D and 3-D Projection

- By adding a third antenna to the InSAR system, Unambiguous Range Magnification (URM) is possible, with the URM ratio being the ratio of baselines. Since there are 3 pairs of satellites now, there would be 2 independent URM ratios (and hence 2 independent slopes).

- 2D Projection is a technique based on the underlying geometry of the satellites to reduce the noise in the system. It is able to reduce the effects of noise on the unwrapped phases.

- Although Projection is a by-product of using URM, the increased accuracy of the unwrapped phases is due to projection, NOT the magnified range itself. Using the

Weighted Least Square algorithm, there is no difference in unwrapping phases that are wrapped 2π or unwrapping phases that are wrapped $URM \cdot 2\pi$.

- 3D Projection takes into account the consistency offered by all 3 satellites, and thus achieves even better results than 2D Projection.

- It is important to take the boundary conditions of the line segments into account. Specifically, one must replicate all line segments in the original 2π by 2π box in 2-D Projection, or 2π by 2π by 2π cube in 3-D Projection, and copy them all around the original box or cube.
- Multiple frequency InSAR has the same underlying principle as multiple baseline InSAR, since the framework of equations is the same. Using multiple frequencies in tandem with multiple baselines could give us far more degrees of freedom.

- The problem of non-collinearity of satellites and the problem of non-integer URM ratios is really due to the same underlying principle – that line segments lie too closely to one another, thus eating into our noise budget. Thus, if we can resolve one, we automatically take care of the other.

Projection used in a 3-satellite cartwheel configuration

- We can obtain all URM ratios in the set $[2,\infty)$ by rotating the satellites in the cartwheel configuration from 0 to 30 degrees.

- Simply mapping all URM ratios to the set of $\{2.1,2.2,2.3,...3.1,3.2,...\}$gives us better results than not doing the mapping at all.

- 3-D Projection fares much, much better in almost all noisy cases across all tilt angles. Specifically, at tilt angles close to 30 degrees, the improvement is the most significant.

- Mathematically, 2-D Projection is simply a special case of 3-D Projection. As such, we can reap all benefits of 2-D Projection using 3-D Projection, and so there really is no real need for 2-D Projection in the first place.

7.2 Flow chart of multibaseline phase unwrapping

The following illustration shows how the different algorithms and processing steps fit together. The numbers in parentheses refer to the chapters in this thesis in which those steps are described in further detail.

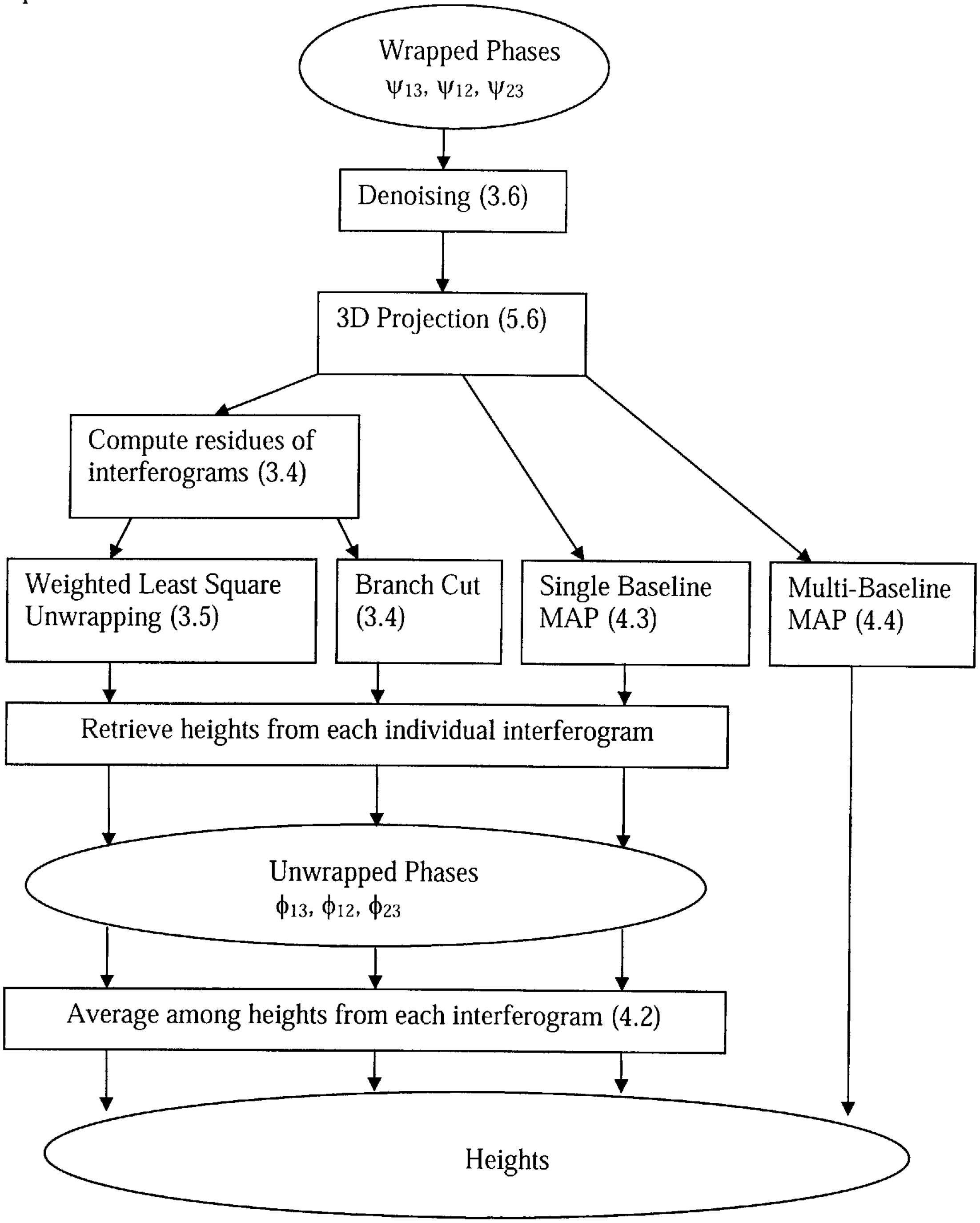

Wrapped Phases
ψ_{13}, ψ_{12}, ψ_{23}
Denoising (3.6)
3D Projection (5.6)
Compute residues of interferograms (3.4)
Weighted Least Square Unwrapping (3.5)
Branch Cut (3.4)
Single Baseline MAP (4.3)
Multi-Baseline MAP (4.4)
Retrieve heights from each individual interferogram
Unwrapped Phases
ϕ_{13}, ϕ_{12}, ϕ_{23}
Average among heights from each interferogram (4.2)
Heights

Figure 7.1: Flow chart of the phase unwrapping and height retrieval process when multiple baselines are available.

8. References

Journal Articles

[1] D.C. Ghalia, L.A. Romero, "Robust two-dimensional Weighted and Unweighted Phase Unwrapping that uses fast transforms and iterative methods," *Journal of the Optical Society of America,* vol. 11, pp 107–17, Jan 1994.

[2] F. Lombardini, "Absolute Phase Retrieval in a Three-Element Synthetic Aperture Radar Interferometer," *CIE International Conference of Radar, Proceedings,* pp 309–312, Oct 1996.

[3] K. Schmitt, W. Wiesbeck, "An Interferometric SAR Processor Avoiding Phase Ambiguities," *International Geoscience and Remote Sensing Symposium,* vol. 4, pp 1713–1715., Aug 1997.

[4] B. Hunt, "Matrix Formulation of the reconstruction of phase values from phase differences," *American Journal of Optical Society,* vol. 69, pp 394–399, 1979.

[5] E. Trouve, J.M. Nicholas, H. Maitre, "Improving Phase Unwrapping Techniques by the Use of Local Frequency Estimates," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 36, pp 1963–72, Nov 1999.

[6] G. Fornaro, G. Francheschetti, R. Lanari, D. Rossi, M. Tesauro, "InSAR Phase Unwrapping using the Finite Element Method," *IEE Proceedings – Radar and Sonar Navigation,* vol. 144, pp 266–74, Oct 1997.

[7] V. Pascazio, "Multifrequency InSAR Height Reconstruction through Maximum Likelihood Estimation of Local Planes Parameters," *IEEE Transactions on Image Processing,* pp 1478–89.

[8] M.D. Pritt, "Multigrid Phase Unwrapping for Interferometric SAR," *International Geoscience and Remote Sensing Symposium,* vol. 34, pp 562–4, May 1996.

[9] G. Fornaro, A. Pauciullo, "Phase Difference-Based Multichannel Phase Unwrapping," *IEEE Transactions on Image Processing,* vol. 14, No. 7, pp 960–72, July 2005.

[10] H.A. Zebker, Y. Lu, "Phase Unwrapping algorithms for radar interferometry: residue-cut, least-squares, and synthesis algorithms," *American Journal of Optical Society,* vol. 15, pp 586–98, March 1998.

[11] R. Goldstein, H.A. Zebker, C.L. Werner, "Satellite Radar Interferometry: Two-Dimensional Phase Unwrapping," *Radio Science,* vol. 23, No. 4, pp 713–20, July–August 1988.

[12] W. Xu, E.C. Chang, L.K. Kwoh, H. Lim, W.C.A. Heng, "Phase Unwrapping of SAR Interferograms with Multi-frequency or Multi-baseline," *International Geoscience and Remote Sensing Symposium,* pp 730-32, 1994.

[13] W. Wong, B.-I. Wu, T. Grzegorcyck, J.A. Kong, "Advances in Interferometric Synthetic Aperture Radar (InSAR) Height Retrieval Techniques," *Preliminary Report,* Nov 30, 2004.

[14] W. Wong, "Synthetic Aperture Radar Interferometry with 3 satellites," Thesis for Master of Science, MIT, June 2005.

[15] H. Xu, Y. Zhou, C. Li, "Analysis and Simulation of Spaceborne SAR Interferometric Baseline," *Proceedings for CIE International Conference on Radars,* pp 639-643, 15-18 October 2001.

[16] A. Liseno, K.P. Papathanassiou, A. Moreira, P. Pierri, "Analysis of forest-slab height inversion from multibaseline SAR data," *International Geoscience and Remote Sensing Symposium,* pp 2660-3, 2005.

[17] G. Ferriauolo, V. Pascazio, G. Schirinzi, "Digital Elevation Model Enhancement from Multiple Interferograms," *International Geoscience and Remote Sensing Symposium,* pp 5471-4, 2005.

[18] F. Zan, G. Fornaro, A. Monti Guarnieri, A. Pauciullo, "Coherent Techniques for Multi Channel SAR Interferometry," *International Geoscience and Remote Sensing Symposium,* pp 1991-4, 2005.

[19] M. Eineder, N. Adam, "A Maximum-Likelihood Estimator to Simultaneously Unwrap, Geocode, and Fuse SAR Interferograms From Different Viewing Geometries into One Digital Elevation Model," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 43, No. 1, pp 24-36, January 2005.

[20] D. Thompson, A.E. Robertson, D.V. Arnold, D.G. Long, "Multi-Baseline Interferometric SAR for Iterative Height Estimation," *International Geoscience and Remote Sensing Symposium,* pp 251-3, 1999.

[21] L. Ying, D.C. Munson, R. Koetter, B.J. Frey, "Multibaseline InSAR Terrain Elevation Estimation: A Dynamic Programming Approach," *IEEE International Conference on Image Processing,* pp 157-160, 2003.

[22] G. Corsini, F. Lombardini, "Simulated Analysis and Optimization of a Three-Antenna Airborne InSAR System for Topographic Mapping," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 37, No. 5, pp 2518-29, Sept 2005.

[23] M.V. Vinogradov, I.V. Elizavetin, “Phase Unwrapping Method for the Multifrequency and Multibaseline Interferometry,” *International Geoscience and Remote Sensing Symposium*, pp 1103–05, 1998.

[24] B.-I. Wu, Y. Hara, Y. E. Yang, and J. A. Kong, “A weighted least squares phase unwrapping technique using residues for SAR interferometry,” *Proc. Progress in Electromagnetics Research Symp. (PIERS)*, Cambridge, p. 717, July 5–14, 2000.

[25] H. Braunisch, B.-I. Wu, and J. A. Kong, "Phase unwrapping of SAR interferograms after wavelet denoising," *Proc. IEEE Int. Geosci. Remote Sensing Symp. (IGARSS)*, Honolulu, vol. 2, pp. 752–754, July 24–28, 2000.

[26] W. Wong, B.-I. Wu, T. Grzegorcyck, J.A. Kong, “Advances in Interferometric Synthetic Aperture Radar (InSAR) Height Retrieval and Coherent Change Detection (CCD),” *Final Report*, Feb 28, 2005.

[27] L. Ying, B. J. Frey, R. Koetter, D.C.Munson Jr., “Analysis of an Iterative Dynamic Programming Approach to 2-D Phase Unwrapping,” *IEEE International Conference on Image Processing*, pp. 829–832, Sept 2002.

[28] F. Gatelli, A. M. Guarniei, F. Parizzi, P. Pasquali, C. Prati, F. Rocca, “The Wavenumber Shift in SAR Interferometry,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, No. 4, pp. 855–865, July 1994.

[29] F. Gini, A. Farina, “Vector Subspace Detection in Compound-Gaussian Clutter Part I: Survey and New Results,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, No. 4, pp. 1295–1311, Oct 2002.

[30] A. Reigber, L. Ferro-Famil, “Interference Suppression in Synthesized SAR Images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 2, No. 1, pp. 45–49, Jan 2005.

[31] Y. Cao, H. Sun, X. Xu, “An Unsupervised Segmentation Method Based on MPM for SAR Images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 2, No. 1, pp. 55–58, Jan 2005.

[32] Z. Li, H. Wang, T. Su, Z. Bao, “Generation of Wide-Swath and High-Resolution SAR Images From Multichannel Small Spaceborne SAR Systems,” *IEEE Geoscience and Remote Sensing Letters*, vol. 2, No. 1, pp. 82–86, Jan 2005.

[33] D. Corr, A. Rodrigues, “Change detection using interferometric SAR data,” Proc *SPIE International Society Opt Eng*, vol. 3869, pp. 127–138, Oct 1999.

[34] A. Wilkinson, M. Datcu, “Interferometric SAR Topographic Mapping, A Bayesian Model-Based Approach,” *Proceedings of Communications and Signal Processing*, pp. 83–88, Sept 1997.

[35] A. Wilkinson, "Synthetic Aperture Radar Interferometry: A Model for the Joint Statistics in Layover Areas," *IEEE International Geoscience and Remote Sensing Symposium 1999*, vol. 5, pp. 2392–2394, 28 June–2 July 1999.

[36] H. Kimura, "A Method To Estimate Baseline And Platform Altitude for SAR Interferometry," *IEEE International Geoscience and Remote Sensing Symposium 1995*, vol. 1, pp. 199–201, 10–14 July 1995.

Books

[37] A. Hein, *Processing of SAR Data: Fundamentals, Signal Processing, Interferometry.* Germany: Springer-Verlag Berlin Heidelberg, 2004, pp.1–20, 99–119, 197–238.

[38] J. C. Curlander, R. N. McDonough, *Synthetic Aperture Radar: Systems and Signal Processing.* John Wiley and Sons, Inc., 1991, pp. 1–66.

[39] G. Franceschetti, R. Lanari, *Synthetic Aperture Radar Processing.* Florida: CRC Press, 2000, pp. 1–65.

[40] S. M. Kay, *Fundamentals of Statistical Signal Processing I: Estimation Theory.* USA: Prentice-Hall, Inc., 1993, pp.27–82, 157–218, 493–566.

[41] M. D. Springer, *The Algebra of Random Variables.* USA: John Wiley and Sons, Inc., 1979.

[42] K. S. Miller, *Multidimensional Gaussian Distributions.* USA: John Wiley and Sons, Inc., 1964.

[43] J. A. Kong, *Electromagnetic Wave Theory*, EMW Publishing, 2005.