Investigation of Developments in Interferometric Synthetic Aperture Radar until 1994

I.G. Hassenpflug, BSc(Eng)

26 September 1996

Thesis submitted to the
Department of Electrical Engineering
of the University of Cape Town in fulfillment of
the requirements for the
Degree of Master of Science in Engineering

Abstract

This thesis examines the topic of Synthetic Aperture Radar Interferometry in a historical perspective, tracing its development from its beginnings in the 1960s up until May 1994. Applications are listed and airborne and spaceborne implementations reviewed.

The underlying theory of interferometry is explained, including a discussion of error sources, and a simulation for point targets is documented to illustrate the interferometric processing steps. The application of the SASAR VHF SAR system to interferometric operation is examined analytically.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank Professor Michael Inggs, my supervisor, for suggesting the thesis topic, for providing valuable support, and for showing me the way ahead. The numerous members of the UCT Radar Remote Sensing Group contributed their time and knowledge: I am especially indebted to Andrew Wilkinson for his invaluable help and encouragement, for the numerous fruitful discussions on interferometry, and for allowing me to adopt his notation conventions. I am furthermore grateful to Jasper Horrell for his patient explanations of SAR processing, and to Peter Golda for the use of his radar simulator and the willingness to explain how it worked. Rolf Lengenfelder and Richard Lord kindly accommodated my requests for computer time.

I also wish to extend my gratitude to the library staff of the University of Cape Town for their help during the literature review for this thesis.

Terms of Reference

The primary objective of this thesis is to document developments in Interferometric Synthetic Aperture Radar (InSAR) from the time of its inception in the 1960s until May 1994. Secondary objectives consist of simulating across track InSAR for point targets, and investigating the suitability of the locally developed VHF SAR system for InSAR operations.

Synopsis

Chapter 1 introduces interferometry in the context of radar remote sensing applications. SAR is a popular tool for radar remote sensing, and one of its applications is in two-dimensional terrain mapping. To obtain three-dimensional terrain images, terrain coherence images, velocity field images or images showing temporal terrain changes, interferometry has been used in conjunction with SAR, resulting in the technique known as Interferometric SAR. Different methods of obtaining topographic data are compared, and applications of InSAR in various fields are described.

Chapter 2 presents the theory of interferometry and a brief summary of SAR theory. The process of obtaining SAR images from coherent radar returns is described and problems of SAR systems such as layover, speckle and motion compensation are noted. Turning to interferometry, the basic geometrical radar-target relationship is described, and the interferometric processing stages detailed. These include image acquisition, image registration, interferogram formation and, subsequently, phase unwrapping and phase conversion. The error sources in InSAR systems are listed, and the sensitivity of the phase measurements to system and terrain parameters discussed. Appendix A contains a simulation of InSAR for point targets which was done to verify the understanding of InSAR theory. The MathCAD worksheets for the simulation are found in Appendix B, the simulation results are found in Appendix C, while the simulation program listings and command files are collected in Appendix D.

Chapter 3 is a review of InSAR implementations, their use, results and shortcomings. Since InSAR is essentially a processing method, implementations considered include dedicated InSAR platforms as well as normal SAR systems whose data has been used in interferometric processing. The InSAR systems considered cover both spaceborne and airborne, single pass and repeat pass, and multi-frequency systems. Since May 1994 a number of InSAR

systems have been proposed, and many studies of interferometry carried out. The thesis however considers only those systems in use by May 1994. Along track interferometry is a method for measuring closing velocities and is used to measure ocean surface current velocities. Across track InSAR is used to generate of topographic maps, coherence and correlation studies of terrain, and for surface change measurement.

Chapter 4 is an analysis of the locally designed VHF SAR system known as SASAR, to investigate its suitability for InSAR operation. The optimal InSAR system parameters are determined and the system's ability to map different terrain slopes given its swath dimensions investigated. The system can map slopes of up to 12° at its highest operating altitude. The sensitivity of the interferometric phase to system and terrain parameters such as baseline length, baseline orientation and terrain topography is examined. The MathCAD worksheets used for the analysis are collected in Appendix E.

Chapter 5 suggests topics of further research into InSAR. Suggestions are the extension of simulations from point targets to distributed surfaces using an improved radar simulator, the implementation of algorithms to convert interferograms to height data in a robust manner, and the investigation of decorrelation effects in InSAR images owing to terrain nature and topography. At the same time data suitable for InSAR processing is required for the development of robust algorithms. This would enable an InSAR simulation program to be a useful tool for projecting the performance of a given InSAR system for a given mission over a specified type of terrain. The SASAR system analysis was successful, and the suggestion is that flat terrain should be mapped first to allow easy processing.

Glossary

- **across track** : direction normal to radar platform motion; if no squint angle, will be equal to slant range or line of sight direction
- **AIRSAR** : Airborne SAR system operated by NASA/JPL
- **along track** : direction parallel to radar platform motion, also known as azimuth direction
- **azimuth direction** : see along track
- **backscatter** : reflected radar energy
- **baseline** : distance between the two antennas in an InSAR system
- **C-Band** : electromagnetic emissions in frequency band 4 to 8 GHz
- **Caltech** : California Institute of Technology
- **CCRS** : Canada Centre for Remote Sensing
- **coherent** : dependent on phase
- **CV-990** : first InSAR system operated by NASA/JPL
- **depression angle** : angle between the horizontal and the radar beam at the antenna
- **DRA** : Defense Research Agency
- **ECCM** : electronic counter counter measures
- **elevation angle** : angle between the horizontal and the radar beam at the antenna, or between downward vertical and radar beam

- **ESA** : European Space Agency
- **ESAR** : European SAR project
- **geometric decorrelation** : cause of phase noise in an InSAR image due to different viewing angles
- **GPS** : Global Positioning System
- **grazing angle** : angle between the horizontal and the radar beam at the target
- **ground range swath** : slant range swath converted to ground distance
- **IMU** : Inertial Motion Unit
- **incidence angle** : angle between the normal to the target surface, and the radar beam at the target
- **INS** : Inertial Navigation System
- **InSAR** : Interferometric Synthetic Aperture Radar
- **JPL** : Jet Propulsion Laboratory of the California Institute of Technology
- **Ka-Band** : electromagnetic emissions in frequency band 27 to 40 GHz
- **Ku-Band** : electromagnetic emissions in frequency band 12 to 18 GHz
- **line of sight (los)** : slant range direction
- **look angle** : angle between the downward vertical and the radar beam at the antenna; also sometimes known as elevation angle
- **multilook image** : multiple look image making use of excess data in the beamwidth
- **nadir** : point directly under the radar platform, where SAR cannot be used
- **NASA** : National Aeronautical and Space Administration

- **P-Band** : electromagnetic emissions in frequency band around 400 MHz
- **PRF** : Pulse Repetition Frequency
- **RAF** : Royal Air Force
- **range direction** : see across track
- **RRSG** : Radar Remote Sensing Group, UCT
- **SAR** : Synthetic Aperture Radar
- **SASAR** : South African SAR project
- **speckle** : multiplicative noise found in coherent images
- **slant range swath** : difference between minimum and maximum slant ranges mapped by the radar
- **thermal decorrelation** : cause of phase noise in InSAR images due to the SNR
- **TOPSAR** : Topographic SAR system operated by NASA and JPL
- **UCT** : University of Cape Town
- **USGS Maps** : US Geological Survey Maps

List of Symbols

Symbol	Explanation
ψ	principal or wrapped interferometric phase difference
φ	absolute or unwrapped interferometric phase difference
φ_{hor}	interferometric phase function for flat terrain
b	interferometer baseline
b_c	critical baseline
$b_\perp$	baseline perpendicular to line of sight direction
b_h	horizontal baseline
θ_b	baseline elevation anlge above the horizontal
θ_1	incidence angle of signal to interferometer baseline
θ_{10}	θ_1 at mid swath
α_1	antenna look angle
α_{10}	look angle at mid swath
γ_1	local terrain incidence angle
γ_{10}	local incidence angle at mid swath
γ_f	incidence angle for flat terrain (α_1)
$\Delta\gamma$	difference in incidence angles of the two antennas
β	surface slope
r_1	slant range to target from antenna 1
r_2	slant range to target from antenna 2
Δr	signal path difference
r_{10}	slant range from antenna 1 at mid swath
θ	look angle to target
N_l	number of coherent looks
S_s	slant range swath width
n_R	slant range sampling distance (bin size)

Symbol	Explanation
λ	radar wavelength
c	speed of light
θ_B	antenna beam width
B	range band width
B_D	doppler band width
δR_s	slant range resolution
δR_g	ground range resolution
δA	azimuth resolution
l	real antenna length
L	synthetic aperture length
v	radar platform velocity
τ	coherence time
γ	correlation coefficient
ρ	decorrelation coefficient
f_0	radar centre frequency
PRF	pulse repetition frequency
SNR	signal to noise ratio
r_u	unambiguous range
$r_{\min}$	range at near swath
$r_{\max}$	range at far swath
k_b	Bragg wave number
k_r	radar wave number
g	gravitational acceleration
ω	angular frequency
$\mathbf{u}$	surface velocity vector
u_x	azimuthal component of surface velocity vector
u_y	across-track component of surface velocity vector
u_r	radial component of surfave velocity vector
$\mathbf{v_1}$	received signal at antenna 1
$\mathbf{v_2}$	received signal at antenna 2
Δr_c	slant range phase cycle (fringe) spacing
Q	factor distinguishing single and repeat-pass InSARs
R_E	radius of the Earth
z	target height in cartesian coordinates

Symbol	Explanation
R	target height in polar coordinates
y	target ground range in cartesian coordinates
ϕ	target ground range position in polar coordinates
z_1	antenna 1 altitude in cartesian coordinates
A_1	antenna 1 altitude in polar coordinates
$\sqrt{\sigma_\varphi}$	phase standard deviation

Chapter 1

Introduction

Radar interferometry is gaining increasing credibility as a technique for rapid, accurate topographic data collection. As DEMs produced in this manner are becoming available to the general science and operational communities, and as a number of interferometric radar systems are in planning and implementation stages, it is important to understand the development and background, accuracy and limitations of the techniques.

1.1 Objective

The objective of this MSc thesis is to document worldwide studies in interferometric synthetic aperture radar (InSAR) for the purposes of generating topographic maps, correlation and coherence studies with application to measuring surface changes over time; to examine the credibility and usefulness of interferometric data generated by various implementations; to explain the basic theory involved; to simulated across track InSAR for point targets; and to carry out an analysis of the locally designed VHF SAR system to determine its suitability for flying InSAR missions.

1.2 Limitations

The objective of the research presented in this thesis limits itself to reviewing and comparing different implementations of InSAR up until May 1994, and identifying important InSAR system performance parameters. The method

used to simulate radar signals has restricted the simulation of InSAR to stationary point targets.

1.3 Outline

This chapter explains the importance of interferometry in the context of radar remote sensing applications, and how to obtain interferometric information using radar. It then continues to give an overview of the thesis.

1.4 Background

Man's efforts are constantly directed towards controlling his environment. The senses are the links to the environment, and in order to exercise control over the environment we need to be able to obtain information. Remote sensing has developed into an important area of scientific research, fulfilling this need for information. It encompasses such tasks as mapping the Earth's surface from satellites, measuring quantities in the atmosphere, on the ocean, on the ground or under it by means of airborne instruments which cover large areas far more rapidly than could be done manually.

1.4.1 Use of SAR in Remote Sensing

A means of remote sensing is electromagnetic waves, the reflected backscatter from surfaces providing information about those surfaces. Different textures of terrain, and different incidence angles of the impinging electromagnetic waves, result in differing amounts of radiation reflecting back to the emitter. Both airborne radar and aerial photography have been widely used for decades [27, 68]. The choice of wavelength depends on the information required. In many cases radar is more suitable than light waves. Radar passes through cloud cover and can be used in all weather, by day and by night, since it provides its own illumination rather than reflected light from the sun. Photography on the other hand requires clear weather [32]. Different wavelengths penetrate differently into the terrain: longer wavelengths penetrate through foliage all the way to the ground, others penetrate to beneath the surface, while shorter wavelengths reflect from the top of vegetation. A commonly used radar system for remote sensing applications is synthetic aperture

radar (SAR). Good azimuth, or along track, resolution can be gained by this method, and it is widely used for mapping.

The image quality of SAR is not as good as that of photographs or real array images, but SAR images are upgradable by scrupulous attention to circuit design and functional design of the radar. Speckle is present because of the use of monochromatic coherent illumination, although this can be alleviated by the use of different carrier frequencies or multilook from different angles [32]. Since the advent of digital signal processing in radar systems, SAR images have been easily portable electronically [32]. Photographic images require chemical development and can only be transmitted through an electrical converter such as a television camera [3]. SAR images are produced using a horizontal offset from the target which makes it unnecessary to overfly the target directly such as is necessary in photography.

However, SAR-generated maps are two-dimensional, whereas in many fields of study three-dimensional imaging is required. Using radar interferometry, a technique in which terrain is viewed more than once, usually from different angles, two or more images yield information which enables the height of the terrain to be determined accurately. Combining radar interferometry with synthetic aperture radar leads to Interferometric SAR (InSAR). InSAR is thus able to collect data rapidly with high spatial and vertical accuracy. A number of experimental InSAR systems have been tested and flown, and there are many different approaches to the data processing aspect of InSAR; these are documented in the literature. InSAR theory has developed substantially from its beginnings in simple geometrical models to more sophisticated and statistical approaches, in step with its development as an application. The use of InSAR has expanded from its initial use in topographic mapping to a more varied acquisition of information about the terrain.

1.4.2 The Use of Phase information in Interferometry

The important difference between SAR images and InSAR is that the phase information of the complex radar return signal is retained throughout the processing [68]. The starting point for InSAR is a set of high quality SAR images. The SAR processor must therefore be one which preserves the inherent phase information in the signal data.

Raney and Vachon [54] discuss the development of a specific phase-preserving SAR processor to support applications such as interferometry, coherent quadra-

ture polarimetry, precision point scatterer calibration and extraction of ocean wave information. The processor deals with range curvature, range walk and azimuth compression in the 2-dimensional frequency domain rather than the range-doppler domain. Multilook is also implemented to reduce the speckle inherent in coherent radar images..

Interference fringes are produced when the two images, resulting from signal reception at two different look angles, are combined point by point to obtain an interferogram whose phase at each point is the difference of the two individual image phases, and whose magnitude is the product of the original magnitudes [68]. The signals are vectorially added, which is mathematically equivalent to multiplying one signal with the complex conjugate of the other. In this manner, the phases are differenced, and the intensities of individual points remain scaled relative to each other.

This section describes in brief the history of research into radar interferometry, from 1974 when the theoretical groundwork was laid [27]. Applications which have been developed (such as mapping), or, have been proposed are described. This chapter is essentially an executive summary of the capabilities of InSAR.

1.5 Sources of Topographic Data

To obtain three-dimensional images, various methods have been experimented with, among them laser altimetry, and photographic or radar stereogrammetry.

1.5.1 Laser Altimeters

Laser altimeters are very accurate for measuring height, but are unsuitable for most topographic purposes since many measurements have to be carried out in two dimensions to cover an area sufficiently for its surface shape to be determined [15].

1.5.2 Stereo-Pair Optical Photography

Vertical relief causes the same terrain to appear in slightly different positions for different look angles. This shift in location, or parallax, is interpreted in terms of height of the terrain. Processing involves the time-consuming

mechanical process of digitization of contour maps [68]. This method is hampered by clouds and poor sunlight conditions which is a particular handicap in the cloud belt within 15° of the equator and at the poles [27].

1.5.3 Stereogrammetry

This involves two passes over the same site at different look angles, and for the track of the radar platform to be well known to register the images. Stereo pair radar imagery is an extension of optical techniques. If a large baseline exists between the two passes over the terrain, the change in slant range to targets changes significantly with the topography and can be measured by correlating the intensities in the two images. There is difficulty in measuring pixel shifts smaller than the image resolution, therefore a large baseline is a requirement, which again is problematic as the target response changes with incidence angle (speckle effect). Processing also involves the time-consuming mechanical process of digitization of contour maps. SIR-B data has been processed, with resultant height errors from 24 m upwards [13, 45].

1.5.4 Shape from Shading

These are methods which use models of terrain reflectance as a function of local incidence angle and terrain type to obtain slope information from radar brightness maps. The brightness map can yield slope information if the dependence of brightness on terrain type and local incidence angle is known. Studies of satisfactory mathematical models are ongoing, and data is being accumulated by airborne and spaceborne SAR sensors [13].

1.5.5 Monopulse Principle applied to SAR

This method works by examining the crosstalk between different polarization return signals from an antenna. If the like- and cross-polarized backscatter is uncorrelated, the antenna crosstalk amplitude and phase is a function of elevation angle only. NASA/JPL tested this method on their AIRSAR system [18].

1.5.6 InSAR

In comparison to radar stereogrammetry, with InSAR a difference in slant range is associated with a small time difference which is expressed as a phase shift [45]. The signal phase is sensitive on the order of a wavelength, thus a smaller baseline is required. This makes the coregistration of images easier, so the method is potentially much more accurate than stereogrammetry. The viewing geometry is shown in Figure 1.1. The positions of antennas 1 and 2 are shown circled, while the target is labelled P. The viewing geometry limits the range of incidence angles which the InSAR system can process, and phase unwrapping remains a problem for all systems [8]. Interferometry has been attempted using airborne and spaceborne SAR systems. These systems differ in several respects and are therefore addressed separately in the following paragraphs.

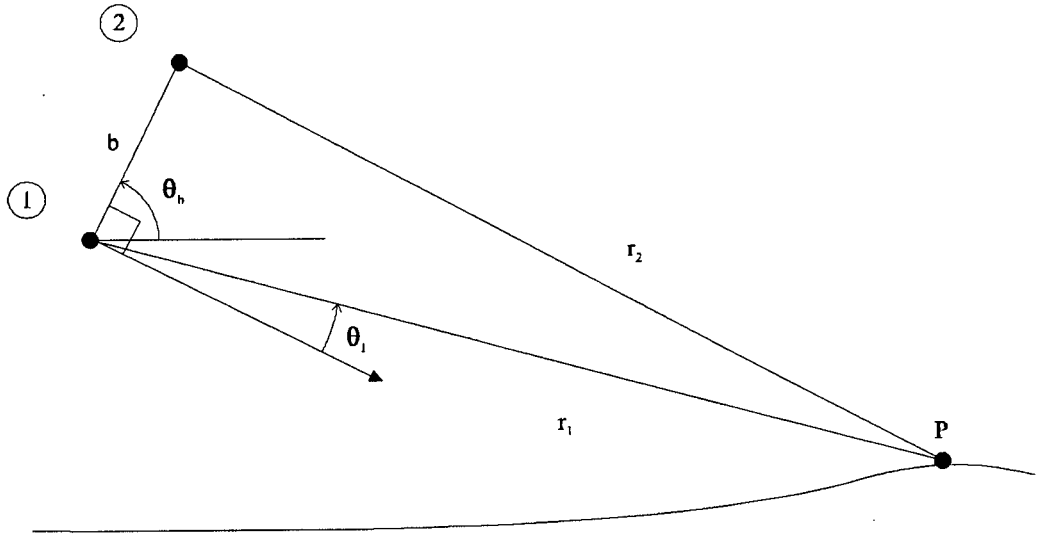


Figure 1.1: Basic across track InSAR Imaging Geometry seen in the along track Direction

Airborne SAR Interferometry

This requires two coherent SAR antennas for one pass systems, or a single antenna used in two pass systems. Displacement of the antennas or flight

paths in the across track direction creates the baseline, and ideally there is no need for ground reference points. The observed terrain shift is of the order of the wavelength used rather than the resolution cell size as in stereo techniques, thus the problem of recognizing resolution elements in two stereo images is circumvented [15]. This is especially difficult in radar images because of speckle: resolution elements may not appear identical in the two images as the views are only statistically related [68]. Range curvature aberrations occur if the SAR system has a high resolution, requiring a modified focussing algorithm.

Spaceborne SAR Interferometry

So far no satellite with two antennas, or tethered satellite systems, are available, and so spaceborne systems are constrained to functioning in repeat pass mode, which limits their use to areas where temporal decorrelation is not rapid [67]. Range curvature aberrations occur if the SAR system is flown at high altitude, requiring a modified focussing algorithm.

1.6 InSAR Applications

Applications of InSAR range from three-dimensional imaging to research of other terrain properties.

1.6.1 Topographic Mapping

The topographic maps can be used for a large variety of geological and geophysical investigations [15]. Furthermore, InSAR can be used for the rectification of the geometry in SAR images and in interpreting radar brightness maps. InSAR becomes a method for decoupling the effects of local topography from other contributions to the radar reflectivity [68].

Topographic maps are useful to prospectors, archeologists and mining companies, as well as to radio network system designers [8].

Earth-based observations of other planets in the solar system have been made using interferometry to separate ambiguous returns from the Northern and Southern hemispheres. The elevation data of the Moon was obtained using interferometry, while changing the baseline due to the Earth's rotation enabled the resolution of ambiguities and obtaining of elevation data from Venus [27].

1.6.2 Differential Radar Interferometry

Three or more images of the terrain are taken, at two together at one time, the third and further ones at differing times. These images can be used to discover surface variations between the images times. Such experiments were carried out after the California earthquake in 1987 [20]. The heaving and buckling of fault zones, plate motions, residual displacements from seismic events, and motion from pre-volcanic swelling have been studied using this technique [28].

1.6.3 Military Applications

Military uses of InSAR include surveillance, reconnaissance, target recognition [51], operations planning and navigation.

1.6.4 Temporal Coherence Studies

Repeat pass systems have been used for temporal coherence studies on a spatial scale not possible with ground based systems and a temporal scale not possible with satellite based systems. Crustal and glacier movements, swelling or shrinking of the soil due to changes in water content have been studied [28].

1.6.5 Correlation Images

ERS-1 data has been used to generate temporal correlation images of terrain. In temperature surveys carried out, a drop below freezing coincides with a change in the dielectric constant of the surface and impacts the complex reflectivity of the surface. Thus two passes over a terrain will be decorrelated if the temperature drops below freezing on only one of the passes. C-Band radar is sensitive to the temperature change for the liquid-solid phase change, thus the decorrelation is quite rapid [70].

1.6.6 Image-based SAR Prediction

If a DEM for a terrain has been obtained, the radar brightness map can be related to the topography of the terrain. The pixel brightness dependence on the local incidence angle and the type of reflecting surface can be determined;

this aids in developing mathematical models to facilitate shape from shading techniques [13].

1.6.7 Ocean Wave Parameters

If the positions from which a point is observed are separated in the azimuth, or along track, direction, then motions on the surface can be measured [25]. Figure 1.2 shows the contrast with across track interferometry. Here the

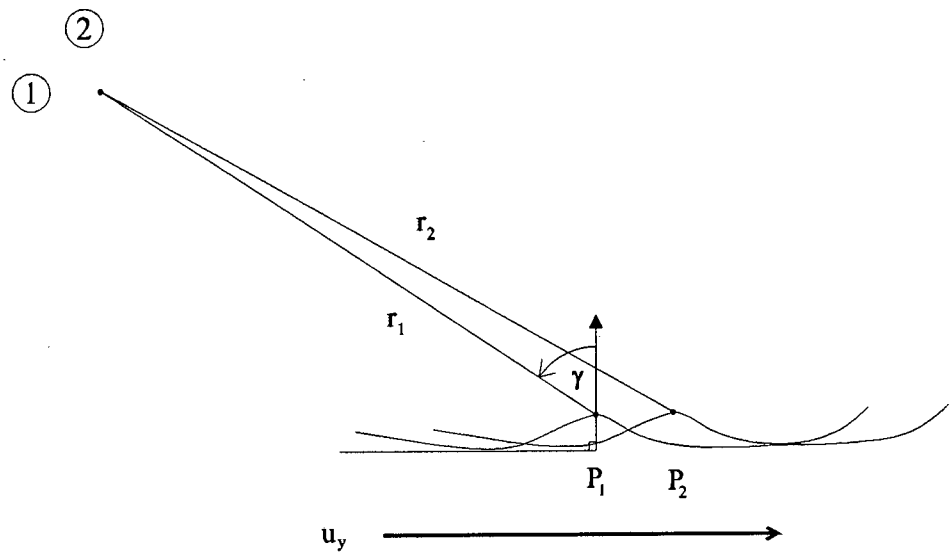


Figure 1.2: Basic along track InSAR Imaging Geometry seen in the along track Direction

antennas occupy the same position but at differing times, while the target shifts position in that time interval. Knowledge of the target radial motion provides instantaneous spatial information about the structure of the ocean wave field [46]. Surface waves can be observed by InSAR as images of the local surface velocity field. The two dimensional velocity field of the ocean is of interest to oceanographers, for wave forecasting, for trans-oceanic ship routing, and for the design of offshore and coastal installations [46]. Line of sight ocean surface currents can be directly measured using along track interferometry [25]. Using more than two baselines in azimuth allows determination of the coherence time of the ocean surface, and InSAR data combined with

backscatter and doppler data are used to understand ocean processes [7]. Measurements of significant wave height in the ocean have also been made by estimating the variance in height over a multi-pixel ocean patch [11].

1.7 Thesis Development

Chapter 2 is a review of SAR theory and interferometry, with the emphasis on InSAR. The principles of SAR and the process of obtaining SAR images are briefly described. The use of SAR for mapping is clarified by explaining how the useful property of SAR, namely good azimuth resolution, in conjunction with the achievement of high range resolution by means of pulse compression, leads to high resolution mapping. The constraints which need to be satisfied in order to achieve the desired resolution and performance of a SAR system are identified. Errors in the processing of SAR images, including motion compensation, are discussed, as well as their effect on the phase information of the resultant images.

Interferometry is then described via the geometrical relationship between the radar platform and target terrain during one or more flypasts. Both along track and across track interferometry are treated, and for the latter type there is a choice of single pass or repeat pass configurations. The processing steps involved in interferometry are discussed. These include image registration, interferogram formation, phase unwrapping, and phase to height or velocity conversion.

Important factors which affect the accuracy of InSAR results are described: these include errors introduced by radar system limitations, errors brought about by the system geometry, errors brought about by the terrain and environment, and data processing errors. Interferometric processing algorithms for motion compensation, image registration and phase unwrapping are considered. The sensitivity of InSAR system phase measurement to the geometrical and terrain parameters is discussed. Appendix A documents an InSAR simulation for point targets which was carried out to test the understanding of the theory of across track interferometry. MathCAD worksheets and command files for the simulation programs are collected in Appendix B. The simulation results are contained in Appendix C, while program listings are found in Appendix D.

Chapter 3 comprises a review of InSAR implementations in Europe and North America. The results of experiments with existing satellite-based SAR

platforms and the growing number of purpose-built airborne InSAR systems are documented, and parameters of the SAR and InSAR systems used are tabulated. Both single pass and dual-pass, and multi-frequency systems are covered. The implementations include along track and across track interferometric configurations.

Chapter 4 consists of an analysis of the locally designed VHF SAR system known as SASAR, to evaluate its suitability for InSAR operation. The important InSAR system parameters are determined from the equations describing the radar-target geometry and the swath dimensions, and the effect of uncertainties in the InSAR system parameters is investigated. The performance of the system for different terrain topography is evaluated and the topographical limitations determined. Appendix E contains the MathCAD worksheets for the analysis.

Chapter 5 comprises the conclusions about the usefulness of the work carried out, and gives recommendations on future research into InSAR, the SASAR system, and on the evolvement of the current InSAR simulation environment.

Chapter 2

Background Theory on SAR and Interferometry

This chapter summarizes the theory of SAR and interferometry, with the emphasis on InSAR. The mathematics and physics of SAR and interferometry are presented, as well as the processing steps required to obtain an interferometric image from two SAR images. Error sources in InSAR systems are discussed, with methods of analyzing and quantifying phase errors. Figure 2.1 shows the process involved in creating interferometric images with SAR systems.

2.1 Synthetic Aperture Radar (SAR)

SAR is a high resolution coherent remote sensor carried either by an aircraft or an orbiting platform. It is the most common type of 'imaging radar', a type of radar that determines the shape of a target [61]. Two important advances in radar and signal processing technique have enabled high resolution radar maps. Firstly, the development of SAR achieves high resolution in azimuth by synthesis of radar data obtained over a much longer aircraft flight path than the azimuth resolution cell. Secondly, pulse compression methods allow good range resolution by transmitting long pulses which are compressed on reception.

To obtain target size or shape requires resolution in range and azimuth. The azimuth, along track, resolution is achieved by using the phase history of the Doppler shifted return signal over a finite integration time in which

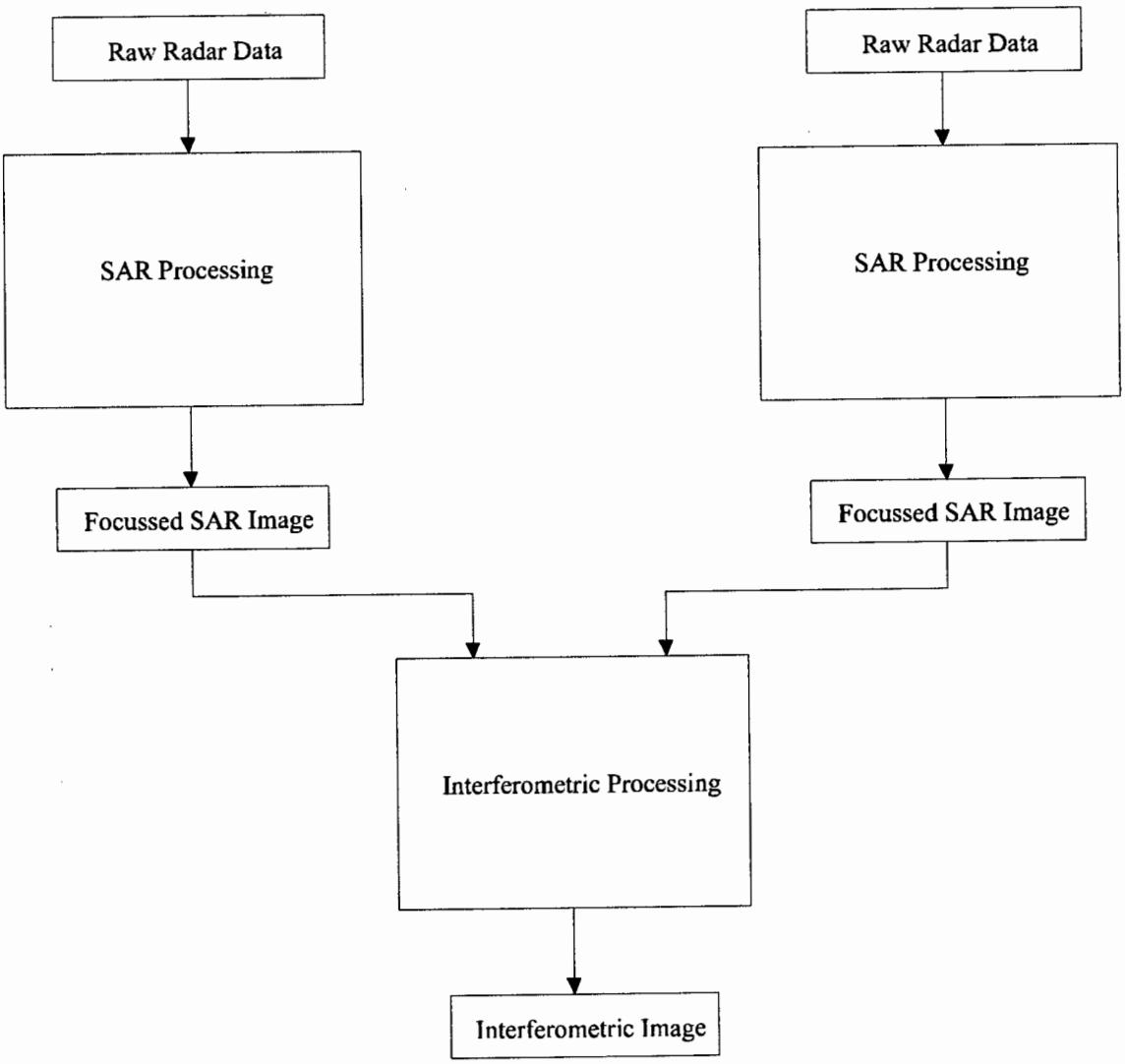


Figure 2.1: Block Diagram of Interferometric Image Production Process

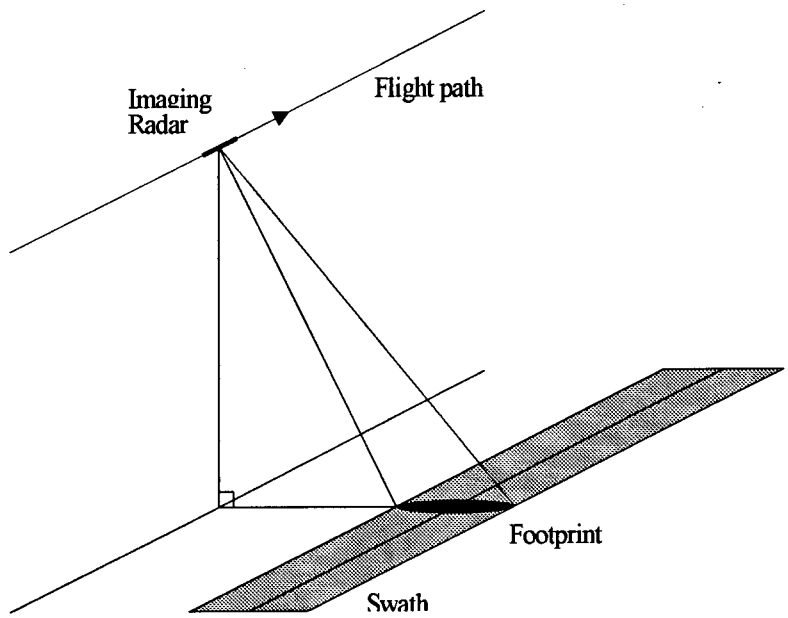


Figure 2.2: SAR Imaging Geometry

the moving radar forms a large synthetic aperture in the azimuth direction, over a considerably longer aircraft flightpath than one azimuth resolution cell [65, 3]. This is done by pointing a small radar antenna out to one side as shown in Figure 2.2, storing the return signals received over a period of time and integrating them to synthesize the equivalent long array antenna. The range, cross-track, resolution is achieved by modulating the transmitted signal and using pulse compression techniques [65, 32].

2.1.1 The SAR Process

Signals received from scatterers arrive at the output of the radar signal processor in phase in order to achieve the narrow beamwidth associated with the long, synthetically generated aperture. First a correction is made for all movement of the aircraft that deviates from the straight-line motion. The motion compensated image is termed the unfocused SAR image [65]. Secondly, the quadratic phase error introduced by the straight-line motion of the radar past each scatterer is corrected. This image is the focussed SAR image [65]. Figure 2.3 shows the steps involved in SAR processing.

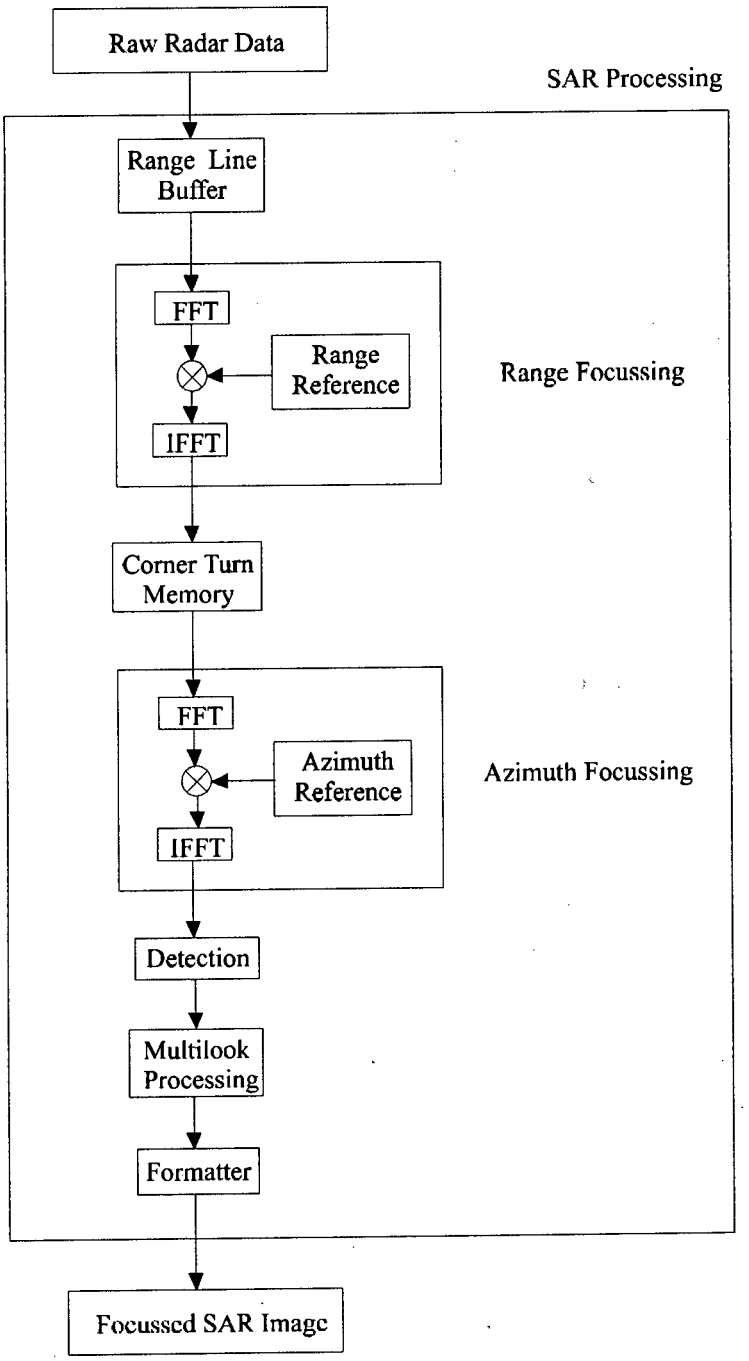


Figure 2.3: SAR Processing

2.1.2 SAR System Parameters

Real Aperture Radar Beamwidth Assuming a uniformly illuminated antenna, the beamwidth of a real array in the far field is derived from diffraction theory as

$$\theta_B = \frac{\lambda}{l}$$

where l is the length of the real antenna [65].

Range Resolution The slant range resolution is given by

$$\delta R_s = \frac{c}{2B}.$$

The ground range resolution is given by

$$\frac{c}{2B} \cdot \frac{1}{\sin \gamma_1},$$

where γ_1 is the incidence angle and B is the transmitter bandwidth [32].

Pulse Compression Range resolution is inversely dependent on the bandwidth of the transmitted pulse [58]. A short pulse has a high bandwidth, giving good range resolution and accuracy, inherent clutter reduction, multipath resolution and a small minimum range [58]. However, the large bandwidth is a disadvantage in that it places stringent requirements on the receiver and transmitter, the large number of range cells may not be able to be displayed on the display screen without overlap, and technical limitations place a limit on the peak power that can be transmitted. This results in limited range [58].

Pulse compression allows the transmission of longer pulses without a reduction in resolution. The pulse length is increased while keeping the bandwidth constant [32]. At the same time the peak power can be increased through the longer transmitted pulse width. A disadvantage is that there is a larger minimum range [32, 58]. The simplest type of pulse compression uses linear frequency modulation (FM) and the pulses are referred to as 'chirp' pulses [61].

Azimuth Resolution The finest azimuth resolution possible for fixed beam SAR is [61, 58]

$$\delta A = \frac{l}{2}$$

This limitation is removed in spotlight mode SAR where the beam is continuously trained on the terrain being mapped [61].

2.1.3 SAR Design Considerations

Maximum Synthetic Aperture Length For coherent integration the target must be in the real beam throughout the formation of the synthetic aperture The maximum synthetic aperture length is

$$L_{\max} = r_1 \theta_B = \frac{\lambda r_1}{l}$$

where r_1 is the range to the target [32, 65]. The smaller the real antenna, the longer the synthetic antenna can be because of the increased available real beamwidth [61].

Resolution and Swath Constraints The resolution and coverage (swath) of a SAR system are interdependent and thus cannot be selected independently. The PRF must be high enough to avoid doppler ambiguities, but low enough to avoid range ambiguities [65].

Minimum PRF The objective is to avoid Doppler ambiguities. The Doppler frequency bandwidth across the real antenna beam is

$$B_D = 2\frac{2v}{\lambda} \sin\left(\frac{\theta_B}{2}\right) = \frac{4v}{l}$$

for an antenna pointed at right angles to the radar platform flight path [32]. To sample above the Nyquist rate requires [65]

$$PRF_{\min} = B_D$$

If $PRF \leq B_D$ aliased targets appear in the image displaced in azimuth from the original target. The aliased targets are of lower intensity and more blurred than the true targets [32]. One pulse must be transmitted each time

the radar platform moves a distance equal to half the real antenna length projected along the radar line of sight. If the spacing between synthetic aperture elements is larger than this, angle ambiguities and image foldover result from grating lobes [32].

Maximum PRF The objective is for unambiguous range at the furthest range in the swath. The unambiguous range is [58]

$$r_u = \frac{c}{2 \cdot PRF}$$

The maximum PRF is [32]

$$PRF_{\max} = \frac{c}{2(r_{\max} - r_{\min})}$$

To incorporate the effects of sidelobe returns and gradual antenna cutoffs, $r_{\min}$ is usually assumed to be 0, so

$$PRF_{\max} = \frac{c}{2r_{\max}}$$

This value is then usually multiplied by a factor of 1.5 to 2.0 to reduce second-go-round returns [32].

Effect of PRF on azimuth resolution The largest possible range of PRF values is [32]

$$\frac{c}{2r_{\max}} \leq PRF \leq \frac{2v}{l}$$

Substituting the limitation on resolution $\delta A \geq \frac{l}{2}$ gives

$$\frac{c}{2r_{\max}} \leq \frac{v}{\delta A}$$

This shows the interdependence of azimuth resolution and closest approach [32].

2.1.4 Image Quality Considerations

Focussing Parameter Estimation To range compress SAR data it is necessary to know the centre frequency of the chirp pulse and the FM slope.

To azimuth compress SAR data it is necessary to know the Doppler centroid (Doppler frequency at the centre of the illuminated swath) and the Doppler frequency slope. Uncertainties in the SAR platform attitude and Doppler echo spectrum require special programs to generate the azimuth reference function. *Clutterlock* estimates the Doppler centroid, while *Autofocus* is a technique to estimate the Doppler frequency slope [65].

Speckle The quality of SAR images is degraded by speckle, or multiplicative noise, which is produced by coherent processing. Speckle arises from random variations in the Earth surface roughness within a resolution cell, manifesting itself as a spatial variation of the pixel intensity. It cannot be reduced by increasing the transmitter power as would be the case for additive, thermal noise. Instead, speckle is reduced by non-coherent integration of two or more independent coherent looks at the same part of the terrain. This is commonly done by summing superimposed processed SAR scenes viewed from different portions of the real radar beam. Multi-looking increases the signal to speckle noise ratio by the square root of the number of independent looks summed [65].

Motion Compensation Angular motions occur in aircraft mounted antennas in the form of yaw, pitch and roll. The antennas must be stabilized and signals motion compensated for deviations from a straight flight path [61]. Gyroscopes stabilize for pitch and roll, while a gyroscope can be used for yaw. Alternatively, clutterlock can be used [32].

To do motion compensation, an acceleration can be measured along the line of sight [61]. Integrating to give velocity and dividing by λ yields the frequency change (doppler) to be applied to the received doppler signals to correct for the measured motion. One way to do this is to vary the receiver signal oscillator. The accuracy of this process must be of the order of the maximum allowed uncompensated acceleration. A further complication is that the line of sight accelerometer is affected by gravity since the region mapped lies below the aircraft. This 'dc' acceleration must first be subtracted in order to obtain the proper compensation [32]. Instead of using Inertial Measurement Unit readings of across track accelerations, it is also possible to use an autofocus algorithm to obtain information about platform motion [4].

Sidelobes Sidelobes are introduced in the storage and filter formation process because of storing signals only to the highest Doppler frequency of interest. They are also introduced by uncompensated non-linear aircraft motion and phase variation in the transmitter and receiver signals through the antenna, transmitter, receiver, and the atmosphere [32]. Peak sidelobes behave as secondary main lobes. A strong target in the direction of a large peak sidelobe is indistinguishable from a return through the main lobe and gives a false target on the display in the direction of the main lobe [32, 61]. Integrated sidelobes have the following effect. If a good integrated power ratio is not obtained, small dark targets in fields of bright targets are washed out [32]. The sidelobes can be reduced at the expense of SNR and a small decrease in azimuth resolution [32].

Range Curvature This refers to the curved response in range that occurs for side-looking SAR as the radar first approaches and then recedes from a point target entering the real antenna beam. The change in range becomes more pronounced at longer ranges and may for spaceborne SAR produce a shift of more than one range cell. This shift is known as range migration and must be removed before azimuth compression is done [65].

Range Walk Range walk results when scatterers enter and leave the beam at different slant ranges. It occurs with spaceborne side-looking SAR because of the Earth's across track rotation beneath the satellite. This must be corrected for before azimuth processing is carried out [65].

2.2 Interferometry

Radar interferometry is a technique using two or more SAR images of the terrain to obtain further information about it. Two configurations exist for InSAR. The first is the single pass, dual antenna, type where two antennas are mounted on the same platform, separated by a fixed baseline. One antenna is used to transmit while both are used to receive the reflected signal. The second type is the repeat pass, single antenna, type where two flypasts over the terrain to be mapped are required.

The InSAR baseline can be created in two ways. If the antenna separation is in the along track direction the interferometer can be used to measure radial

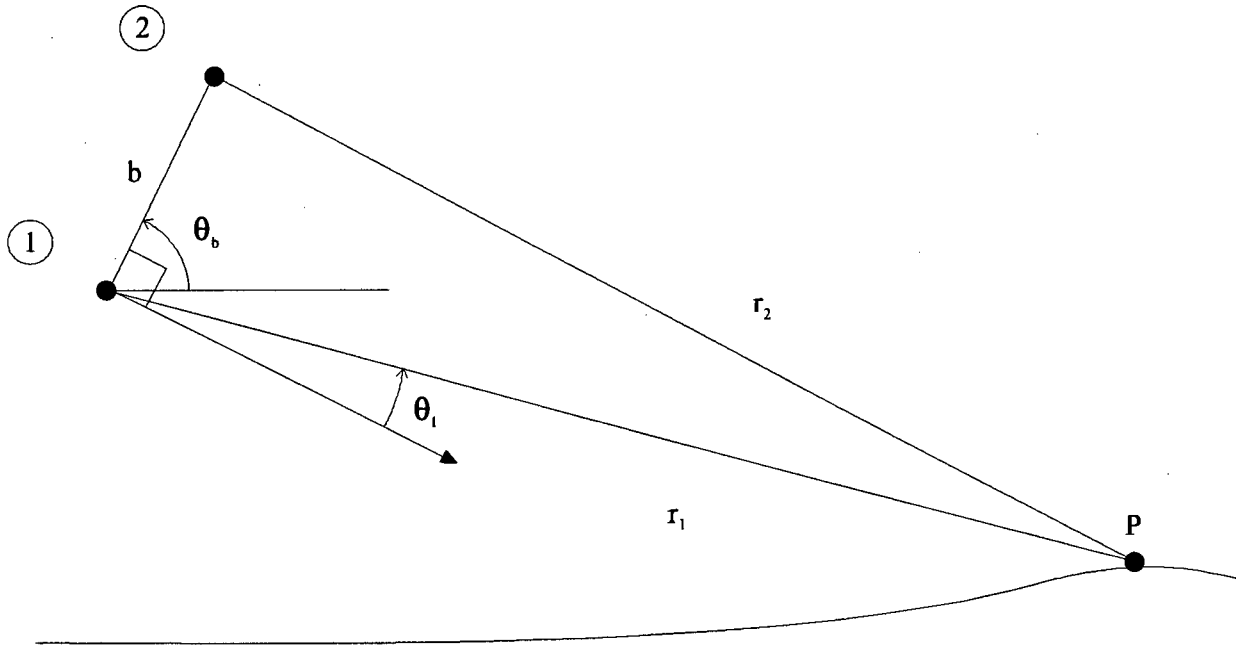


Figure 2.4: Basic across track Imaging Geometry

velocities. If the separation is in the range direction the interferometer can be used to estimate the topography of the terrain.

The steps involved in interferometric processing of the two SAR images are shown in Figure 2.13.

2.2.1 Across Track Interferometry

Graham [27], Zebker [68, 69, 67], Li [39], and Gray [28] derive equations from the radar interferometry geometry to describe the interaction of two images of an area obtained from two slightly displaced locations with either single pass or repeat pass interferometers.

Geometry

The geometric relations derived allow the deduction of true ground ranges and heights.

Single Pass System: A pulse of centre frequency f_0 is transmitted by antenna 1 in the direction of the point target P. The pulse is reflected from P and the echo is received by both antennas 1 and 2.

Repeat Pass System: The radar platform makes two passes over the target area, its position during the two passes designated by points 1 and 2. During the first pass, a pulse of centre frequency f_0 is transmitted by antenna 1 in the direction of the point target P. The pulse is reflected from P and the echo is received by antenna 1. During the second pass, the antenna is in the position marked 2 when it transmits a pulse, and the echo is received by the antenna. If the orbits are non parallel, crossed orbit interferometry results. Gabriel and Goldstein [19] derive a method of obtaining estimates of the orbit parameters from the interferogram.

Equations

Target Reconstruction Equations Thus the absolute phase difference is written as

$$\varphi = \frac{2\pi Q}{\lambda}(r_1 - r_2)$$

where $Q = 1$ for dual antenna single pass systems, and $Q = 2$ for single antenna dual pass systems.

As a function of InSAR parameters [66]:

$$\varphi(\theta_1, r_1, b, \lambda) = \frac{2\pi Q}{\lambda}\left[r_1 - \sqrt{b^2 + r_1^2 - 2r_1 b \sin\theta_1}\right]$$

which after rearrangement gives the angle θ_1 in terms of φ:

$$\theta_1(\varphi, r_1, b, \lambda) = \sin^{-1}\left(\frac{\lambda\varphi}{2\pi Q b} + \frac{b}{2r_1} - \frac{\lambda^2\varphi^2}{8\pi^2 Q^2 b r_1}\right)$$

which is exact for a point target under ideal conditions.

Where $r_1 \gg b$, r_2 is approximately parallel to r_1 the equations simplify to

$$\varphi(\theta_1) \simeq \frac{2\pi Q b \sin\theta_1}{\lambda}$$

$$\theta_1(\varphi) \simeq \sin^{-1}\left(\frac{\lambda\varphi}{2\pi Q b}\right)$$

Topographic Coordinates The measured parameters (r_1, φ) are transformed to polar coordinates (r_1, θ_1) by using the topographic reconstruction equation, and then to Cartesian coordinates (y, z) for mapping information purposes.

Flat Earth Geometry Figure 2.5 shows the geometry of across track InSAR in the flat Earth approximation. Antenna 1 is at (y_1, z_1) and antenna

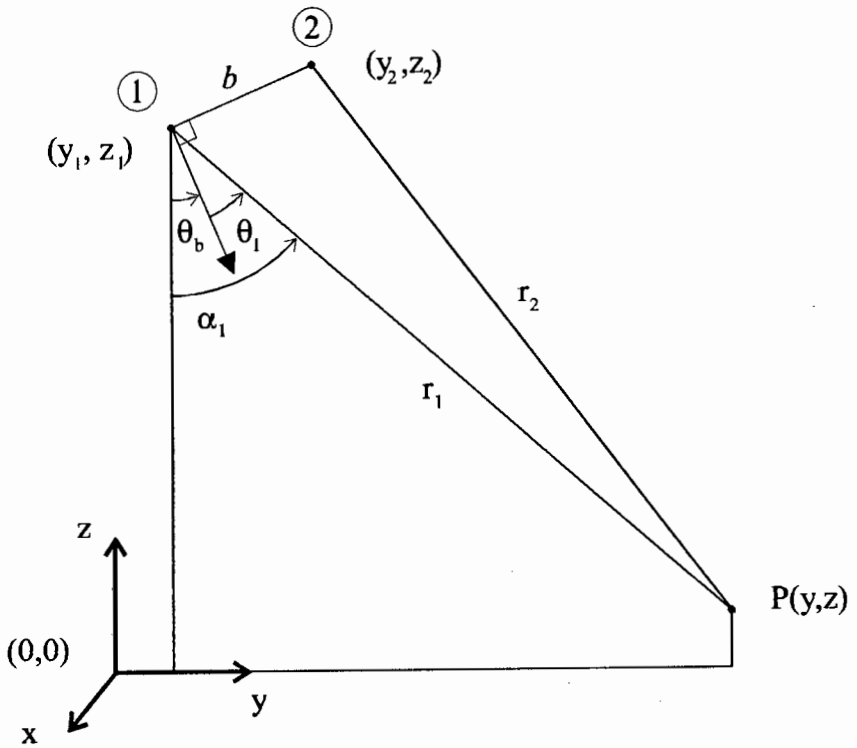


Figure 2.5: InSAR Flat Earth Geometry

2 is at (y_2, z_2). The baseline is elevated at an angle θ_b. The target is located at $P(y, z)$ which is expressed in terms of radar parameters r_1 and α_1 and estimates of b and θ_b. The angle of arrival of the received echo θ_1 is then computed [66].

$$\alpha_1 = \theta_1 + \theta_b$$
$$y = y_1 + r_1 \sin \alpha_1$$
$$z = z_1 - r_1 \cos \alpha_1$$

Spherical Earth Geometry The spherical Earth geometry is shown in Figure 2.6, with all lines in the same plane. Antenna 1 is at polar coordinates (A_1, ϕ_1), while antenna 2 is at coordinates (A_2, ϕ_2), where A_1 and A_2 are the distances from the Earth's centre and ϕ_1 and ϕ_2 are the angles from the reference line.

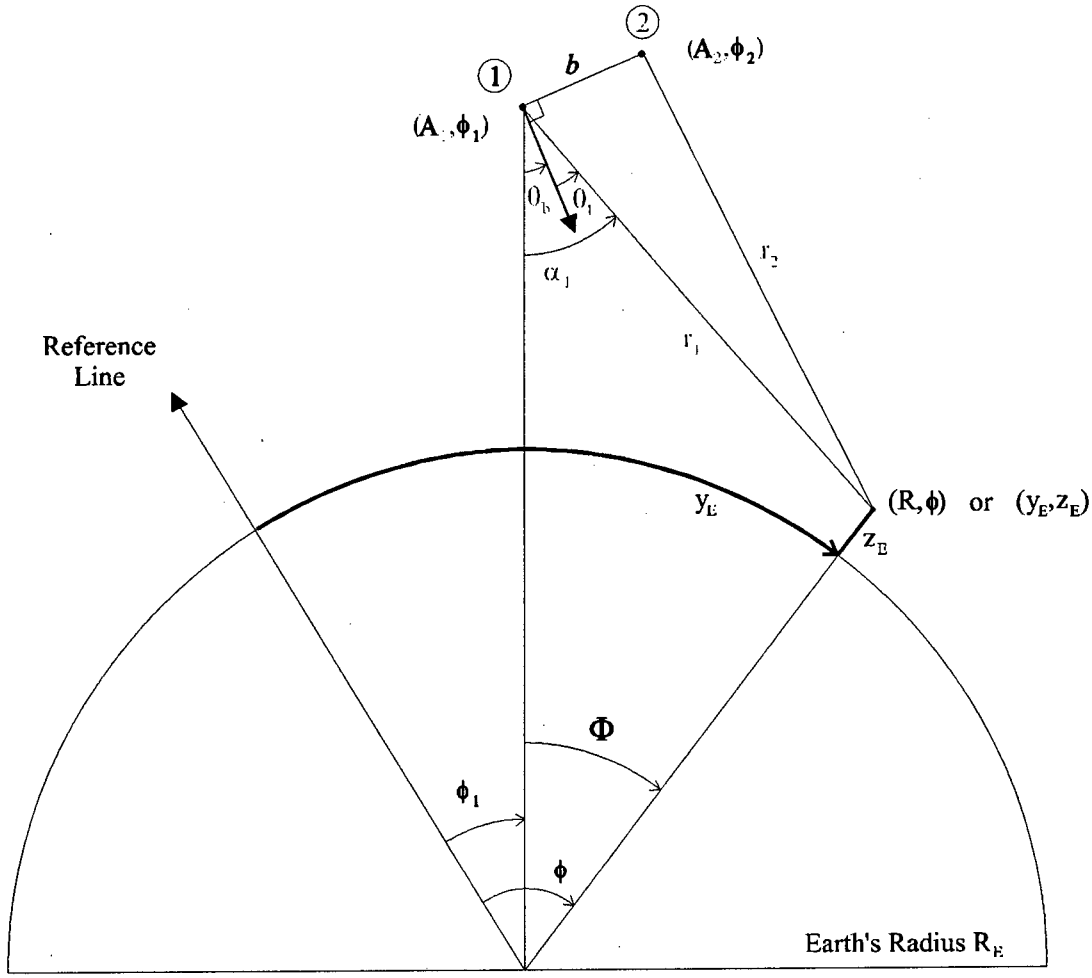


Figure 2.6: InSAR Spherical Earth Geometry

The point target is at (R, ϕ). The topographic information can be expressed as (z_E, ϕ) or (y_E, z_E) where [66]:

z_E = height above reference sphere of radius R_E

y_E = distance along the reference sphere from the reference line

ϕ = angle between reference line and target position

θ_1 is computed from the measured values of r_1 and α_1. The target location in polar coordinates is

$$\alpha_1 = \theta_1 + \theta_b$$

$$R = \sqrt{A_1^2 + r_1^2 - 2A_1 r_1 \cos\alpha_1}$$

$$\phi = \phi_1 + \sin^{-1}\left(\frac{r_1 \sin\alpha_1}{R}\right)$$

The location can be expressed in terms of the surface height and location above the spherical earth model of radius R_E:

$$z_E = R - R_E$$

$$y_E = R_E\phi$$

Various Derivations of Phase to Height Conversions

Different authors [68, 15, 69, 45, 70] use differing symbols and conventions in deriving similar results, some of which are presented here for reference. All derivations assume a flat earth model.

Derivation Number One Zebker and Goldstein [68] use the geometry of Figure 2.7 in their equations for a single pass system. The elevation angle α_1 is used to calculate the radar height above the target

$$z_1 - z = r_1 \sin\alpha_1$$

$$z_1 - z = r_1 \sin(\cos^{-1}\left(\frac{\lambda\varphi}{2\pi b}\right) - \theta_b)$$

$$y = \sqrt{r_1^2 - (z_1 - z)^2}$$

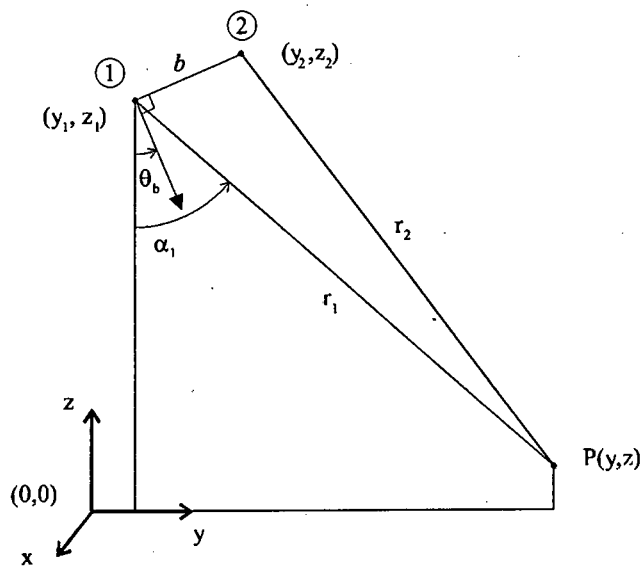


Figure 2.7: InSAR Geometry Derivation 1

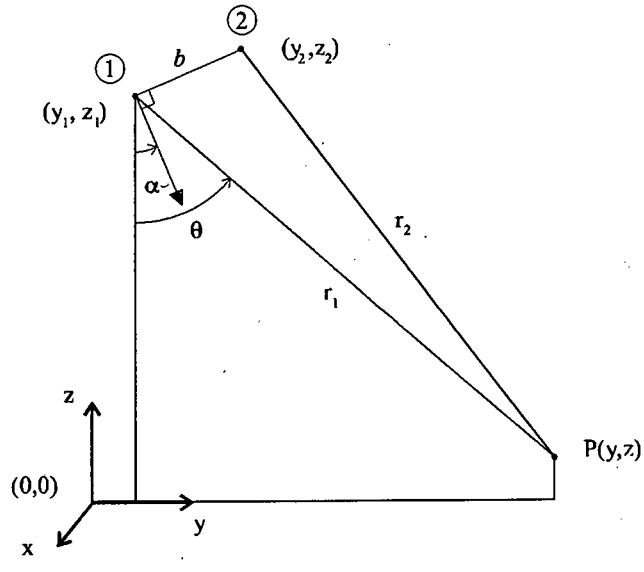


Figure 2.8: InSAR Geometry Derivation 2

Derivation Number Two Zebker [15, 69, 70] uses the geometry of Figure 2.8 in a derivation for a dual-pass system. The terminology for α_1 and θ_b are reversed in this derivation. The path length difference δ is used to obtain the angle between the baseline normal and the look direction, termed θ_1 in the InSAR geometry used in this thesis [66]

$$\delta = r_1 - r_2 = \frac{\phi\lambda}{2\pi}$$

$$\sin(\theta - \alpha) = \frac{(r_1 + \delta)^2 - r_1^2 - b^2}{2r_1 b}$$

$$z = z_1 - r_1 \cos\alpha \cos(\alpha - \theta) + r_1 \sin\alpha \sin(\alpha - \theta)$$

Derivation Number Three Madsen, Zebker and Martin [45] use the geometry of Figure 2.9 in their derivation of the InSAR equations for a single pass system.

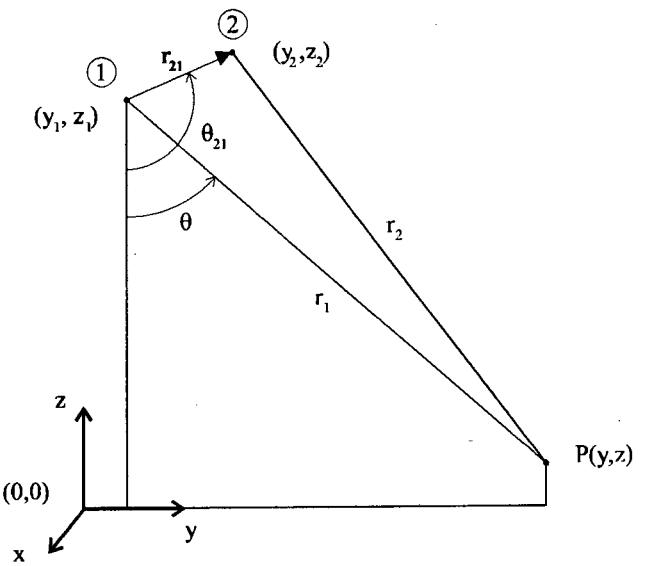


Figure 2.9: InSAR Geometry Derivation 3

$$\varphi = -\frac{2\pi}{\lambda}\Delta r$$

where Δr is the path length difference for signals from the target to the antennas. Therefore

$$\frac{\varphi\lambda}{2\pi} = -\Delta r = |\mathbf{r}_{21}| \cos(\theta - \theta_{21})$$

where $\mathbf{r}_{21}$ is the baseline vector. The look angle is used to obtain the target coordinates:

$$\theta = \theta_{21} - \cos^{-1}\left(\frac{\varphi\lambda}{2\pi\,|\mathbf{r}_{21}|}\right)$$

$$z = z_1 - r_1 \cos\theta$$

$$y = r_1 \sin\theta$$

Derivation Number Four Cumming and Hawkins [11] use the geometry of Figure 2.10 to derive InSAR equations for a single pass system. The path

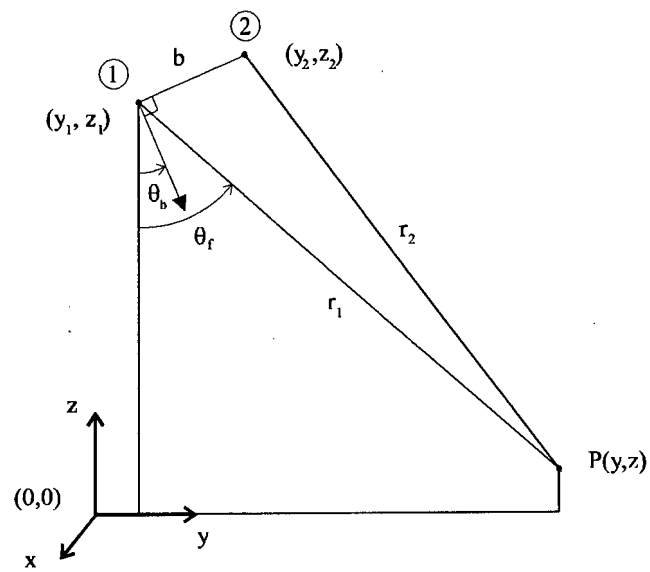


Figure 2.10: InSAR Geometry Derivation 4

length difference for the two antennas is

$$\delta = \frac{\varphi\lambda}{2\pi}$$

This leads to target coordinate equations:

$$\theta_f = \sin^{-1}\left(\frac{\delta}{b}\right) + \theta_b$$

$$z = z_1 - r_1 \cos\theta_f$$

$$y = r_1 \sin\theta_f = (z_1 - z)\tan\theta_f$$

2.2.2 Along Track Interferometry

Along track interferometric SAR is often used to measure the across track surface velocity or the coherence time of the ocean by acquiring two images in identical geometries separated by a small time interval. SAR images of the ocean are a map of surface reflectivity, which is only indirectly related to the complex modulation of the surface reflectivity by longer waves and currents. InSAR directly maps the radial or line of sight surface velocity components. Usually single pass dual antenna systems are used [6, 25], although a proposal has been made for a single pass single antenna system, which will allow InSAR processing of existing SAR datasets [16]. Figure 2.11 illustrates the geometry of along track InSAR, where the target is imaged at slightly different positions by each antenna owing to its range direction velocity component u_y.

Theory

Microwaves interact with ocean surface. For gently undulating surfaces between dielectric and air only one spatial Fourier component of the surface is dominant in reflecting a radar wave. The component must be aligned at right angles to the line of sight direction, and its projected wavelength in this direction must equal odd multiples of half the radar wavelength. This is known as the Bragg condition. Longer waves modulate the shorter Bragg waves, and so the orbital motion of the swell can be seen through this mechanism. The observed velocity, in line of sight direction, seen by InSAR is the vector sum of the swell-induced motion and the Bragg wave phase velocity, plus the surface portion of any ocean currents present. Motion of the surface causes a radial doppler shift according to the phase velocity of the water waves. This doppler shift is observed by the radar [25]. Figure 2.12 shows the radial velocity of surface scatterers imaged by the InSAR system.

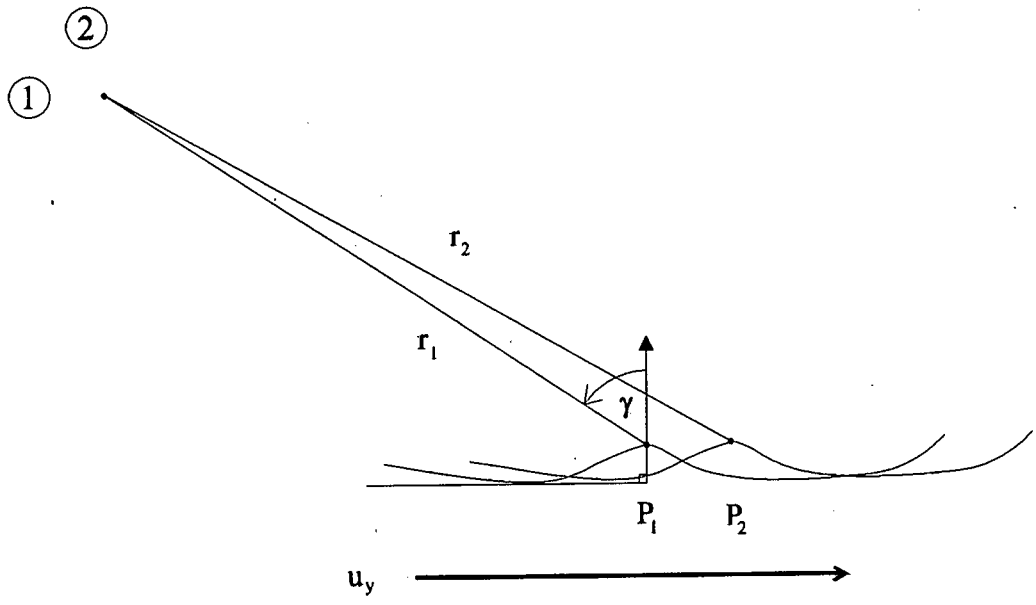


Figure 2.11: Basic Geometry for Along-track InSAR seen in the along-track direction

Equations

As seen in Figure 2.11 the scatterers movement results in a path difference for the radar antennas at each pixel. This results in a phase difference at each pixel in the two SAR images proportional to the radial distance moved by the scatterers [25, 57]. The phase difference translates into a radial one way path difference of

$$\Delta r = \frac{1}{2}\frac{\varphi\lambda}{2\pi}$$

which is the distance moved in the time between observations

$$\Delta t = \frac{b_h}{v}$$

where b_h is the horizontal along track distance between the antenna positions, and v is the radar platform velocity. The radial velocity of the scatterer to result in this path difference is [57]

$$u_r = \frac{\Delta r}{\Delta t} = \frac{\varphi\lambda v}{4\pi b_h}$$

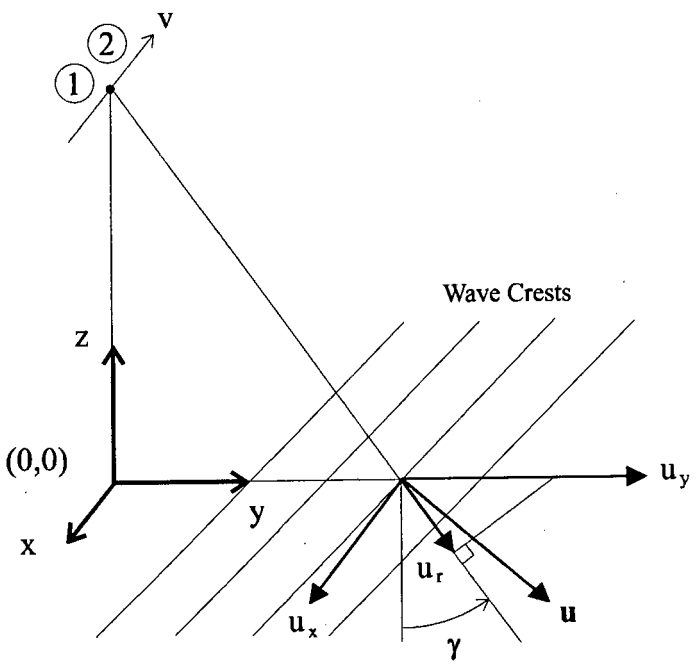


Figure 2.12: Radial Velocity of Surface Scatterers in Along-Track InSAR

The surface velocity u_y is

$$u_y = \frac{u_r}{\sin \gamma_1}$$

as shown in Figure 2.12. Using wavenumber notation, the radar wavenumber is

$$k_r = \frac{2\pi}{\lambda}$$

and the Bragg wavenumber on the surface is [6, 57]

$$k_b = 2k_r \sin \gamma_1$$

which is uniquely defined by λ and the incidence angle γ_1. This lead to the surface velocity expression

$$u_y = \frac{\varphi}{k_b \Delta t}$$

Flying two orthogonal paths allows both orthogonal components u_y and u_x of the current velocity vector $\mathbf{u}$ to be determined.

2.3 InSAR Processing

The implementation of an interferometric processor is a multistage process as seen in Figure 2.1. In this section the basic building blocks of interferometric processing are described.

The steps required for the production of a topographic map using interferometry are:

1. Fly SAR mission(s)
2. Motion compensate and record SAR echoes
3. Form 2 focussed SAR images
4. Register the images accurately
5. Compute the interferogram
6. Resolve phase ambiguities with unwrapping algorithm
7. Use calibration points to compute interferogram phase offset and any unknown system parameters, e.g. baseline length or orientation

8. For each pixel in the unwrapped interferogram compute the location of the three dimensional height map

9. Resample the three dimensional height map onto an evenly spaced grid

10. Apply additional smoothing and other map making functions, e.g. grid registration, mosaicking, symbology and elevation contouring

Figure 2.13 illustrates the interferometric processing steps, while Figure 2.3 illustrates the prior SAR processing steps. The first step in interferometric processing is to obtain two SAR images of the same scene taken from two slightly different look angles. After the application of standard SAR focussing algorithms, the images are accurately aligned to within a fraction of a resolution cell by using an image registration algorithm. Depending on the system type and geometry, complicated image registration may or may not be required. For short baseline, narrow swath systems, linear translations of one image relative to the second may be adequate [66]. For large baseline, wide swath systems, or where the radar platform tracks are non-parallel or cross each other, translation, rotation and stretching may be required to register the images adequately [19].

Once the images are registered, the interferogram of phase differences is obtained by subtracting the phase in one image from the phase in the other on a pixel by pixel basis. The phase resulting from the radar's reflection from a horizontal surface is subtracted from this result, leaving only phase differences corresponding to height variations in the terrain [66]. The interferogram phase $\psi(x,r)$ is only known modulo 2π, i.e. it is the wrapped interferometric phase. Phase unwrapping is the process of resolving the 2π ambiguities. To facilitate the process, the interferogram is usually filtered to reduce phase noise. Also some areas in the interferogram may be corrupted owing to layover, shadowing or decorrelation effects. These areas must be identified and masked out of the unwrapping process [66].

The unwrapping algorithm unwraps the interferometric phase on a relative basis so the unwrapped phase is offset from the required absolute phase $\varphi(x,r)$ by an unknown constant. At least one calibration point of known location on both the interferogram and on the terrain surface is required to evaluate this constant [66].

The constant is then added to each phase value of the unwrapped interferogram. An additional complication results from not knowing some parameters of the interferometric geometry to sufficient accuracy, e.g. baseline

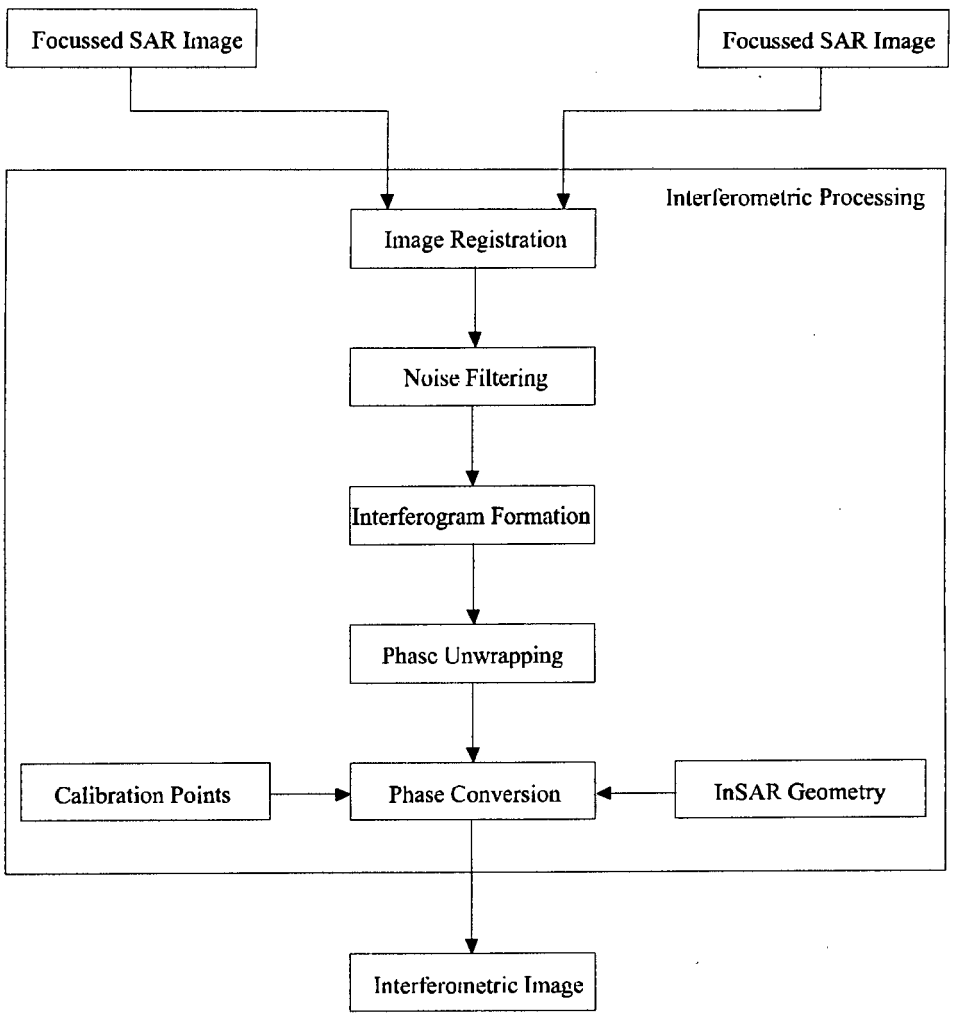


Figure 2.13: Block Diagram of InSAR Processing

length and orientation, and radar height. Additional calibration points are necessary to resolve these parameters accurately [66].

The unwrapped absolute phase interferogram used in conjunction with the knowledge of the interferometric geometry allows a reconstruction of the three dimensional location of each point on the interferogram. The resulting height data set is the resampled and smoothed to obtain a height map, represented in the form of a contour map or some other projection [66].

2.3.1 Focussing

For along track interferometry, Raney [53] describes how focussing and tracking of ocean waves from airborne SAR can be done optimally, so that multiple looks are registered, and wave contrast enhanced. The optimal focus for viewing an ensemble of waves is to focus as though on a non-moving target. Image adjustment is obtained by non-coherent means. A wave image is a brightness mapping of the waves, so the brightness map moves with the waves. Therefore for multilook, each subimage needs to be shifted with respect to the other subimages in accordance to the scene movement between looks. This noncoherent compensation results in registered subimages and the 180° ambiguity of wave direction can be resolved by looking at the shift required to register the looks.

2.3.2 Image Registration

The image registration procedure is critical to InSAR. The images need to be aligned to subpixel accuracy; one tenth of a resolution is considered satisfactory [41]. As misalignment increases, so the SNR of the resulting interferogram decreases (phase noise increases), and errors result in the reconstructed digital elevation model [66].

The two images can be viewed as projections of the three dimensional surface onto two distinct two dimensional coordinate systems. The terrain surface is specified in terms of x, y, z or as height $h(x, y)$. The radar images are projections onto the (x, r) coordinates. As they are taken from slightly different look angles, the projections are slightly distorted with respect to one another. The exact nature of the distortion depends on the flight paths of the antennas and the variation of the terrain surface.

In the case where two antenna orbits are parallel, such as for single pass systems and some dual pass satellite systems, the images have the same

orientation and dimensions in the azimuth direction. Simple translation in this direction will align them in azimuth. In the range direction there is stretching, the differing incidence angles leading to different ground pixel sizes in the two images. Thus one image needs to be stretched in range to fit onto the other. The images also have different range and azimuth pixel offsets, so that one image is usually resampled in order to register the two [41]. If slant range pixel size of the images differs markedly the process of registration by translation and scaling cannot be done independently [41].

Alignment of two images at a particular point can be done by extracting a small patch from one image, centred on the point to be matched to the other image, and then correlating it with the other image until a point of optimum match is found. The criterion is usually chosen to be the SNR of the resulting interferogram phase within the patch, i.e. where the phase noise is at a minimum.

Statistical correlation techniques can be used to register images [41]. The methods involve integer registration and fractional registration. First, images are correlated over discrete pixel offsets and shifted accordingly, then fractional registration is implemented by interpolating the correlation function to find its maximum position [12]. Oversampling of the images is performed by a 2-dimensional cubic spline function [12] or by an 8:1 FFT interpolation [52]. Shifts for each pixel are then calculated using a bilinear interpolation of these shift values., and then cross correlation of the image segments is done to maximize the contrast of the interference fringes in these areas. A variation of this method involves evaluating the spectral intensity of the interferogram on an integer grid and interpolating this by means of a quadratic function to find the maximum position [70].

Lin and Vesecky use the Downhill Simplex method to optimize the registration parameters. The method was tested on SEASAT images [41, 43]. The second image is interpolated on a subpixel basis, the average fluctuation function of the phase difference image is evaluated and the registration parameters (offset and scalings) are adjusted according to the change in the average fluctuation function. The second image is interpolated again and the process repeats until the average fluctuation function reaches a minimum. The criterion used is:

$$f = \sum_i \sum_j \frac{|\psi_{i+1,j} - \psi_{i,j}| + |\psi_{i,j+1} - \psi_{i,j}|}{2}$$

No assumptions are made about the local slope and thus errors would occur

at roll over points in the phase (where the change $2\pi \rightarrow 0$ occurs). By assuming a phase difference between adjacent pixels of less than one half a cycle using the wrapping operator $W()$ [66],

$$F = \sum_i \sum_j \frac{|W(\psi_{i+1,j} - \psi_{i,j})| + |W(\psi_{i,j+1} - \psi_{i,j})|}{2}$$

2.3.3 Interferogram Formation

This section describes the formation and filtering of an interferogram prior to phase unwrapping.

The interferogram is formed by subtracting the phases in two registered SAR images and rearranging the result to lie in the principal phase range of $-\pi < \psi \leq \pi$.

This is done using the wrapping operator $W()$ so that $\psi = W(\psi_1 - \psi_2)$.

Subtraction of the Horizontal Surface Component

The interferogram phase for a flat horizontal surface appears as a set of parallel fringes. The spacing between fringes increases slowly as one moves from near to far range across the swath. Any deviations on the surface show up as deviations of the parallel fringe lines on the interferogram. Because terrain can be modelled as a fluctuating component superimposed on a flat horizontal surface or the curve of the Earth's surface, it is common practice to decouple the smooth horizontal surface component from the fluctuating component. The resulting interferogram more closely resembles a contour map and is therefore easier to interpret visually.

Absolute phase is written as:

$$\varphi(x, r) = \varphi_{hor}(x, r) + \varphi_{fluct}(x, r)$$

To remove the horizontal component from the interferogram we subtract the horizontal component and rewrap the phase [29]:

$$\psi_{fluct}(x, r) = W(\psi(x, r) - \varphi_{horiz}(x, r))$$

Linear Approximation

For a flat horizontal surface model, the phase function can be approximated by a linear phase slope in the range direction:

$$\varphi_{hor}(x_1, r_1) \simeq \varphi_{hor}(x_{10}, r_{10}) + \frac{\partial \varphi}{\partial r_1}|_{r_{10}}(r_1 - r_{10})$$

where the slope is computed at a point at the center of the swath (range r_{10}) [66]:

$$\frac{\partial \varphi}{\partial r_1}|_{r_{10}} = \frac{2\pi Q b \cos\theta_{10}}{\lambda r_{10} \tan\gamma_{10}}$$

Flat Horizontal Subtraction: Exact Calculations

For aircraft imaging systems, the horizontal surface component is modelled as a flat surface. The exact phase function as a function of range r_1 is calculated using [66]:

$$\theta_1 = \cos^{-1}\left(\frac{z_1}{r_1}\right) - \theta_b$$

$$r_2 = \sqrt{b^2 + r_1^2 - 2br_1\sin\theta_1}$$

$$\varphi = \frac{2\pi Q}{\lambda}(r_1 - r_2)$$

Spherical Horizontal Subtraction: Exact Calculations

For spaceborne wide swath imaging systems, the horizontal surface component may be modelled on a spherical surface. The correct phase as a function of range r_1 is calculated using [66]:

$$\theta_1 = \cos^{-1}\left(\frac{A_1^2 + r_1^2 - R_E^2}{2A_1 r_1}\right) - \theta_b$$

$$r_2 = \sqrt{b^2 + r_1^2 - 2br_1\sin\theta_1}$$

$$\varphi = \frac{2\pi Q}{\lambda}(r_1 - r_2)$$

Interferogram Filtering

To facilitate phase unwrapping, it is often necessary to filter the interferogram prior to the application of a phase unwrapping algorithm. The thermal, speckle and phase noise inherent in the interferometric image can be reduced by multilooking, summing over adjacent pixels. The cost is a reduced spatial resolution in the final topographic map. Unlike conventional filtering where the images are smoothed by spatial averaging, the interferogram phase information is still in its wrapped from. Conventional averaging breaks down at the phase roll over points (where the change from $2\pi \rightarrow 0$ occurs). In order to circumvent this problem, vector averaging is usually used [66].

The simplest form of vector averaging is to express each phase value as a complex vector with unit magnitude, add the vectors, then compute the argument of the resulting vector:

$$\hat{\psi} = \arg\{\sum_{i=1}^{N} e^{j\psi_i}\}$$

The summation is made over a small patch centred on the point at which the phase is to be estimated [29]. It has been suggested [39] that the SAR amplitude components be included as weighting factors for each phase value:

$$\hat{\psi} = \arg\{\sum_{i=1}^{N} \mathbf{v}_{1i}^{*}\mathbf{v}_{2i}\} = \arg\{\sum_{i=1}^{N} v_{1i}v_{2i}e^{j\psi_i}\}$$

Including the amplitude factors helps suppress the vector contribution of noise corrupted phase values.

Goldstein used a 2-dimensional triangle function with a phase slope in the across-fringe direction equal to the average fringe rate. This smoothing reduced the number of residues in a sample image from 77500 to 6876, and reduced the spatial resolution from 20 by 20 m to 50 by 50 m [26].

Small et al [59] uses the Cramer-Rao bound condition to choose the number of looks in ERS-1 images required to drive the standard deviation of the phase to less than 30°.

2.3.4 Phase Unwrapping

Unwrapping of phase depends on the system parameters. For across track interferometry, the parallax angle between passes must be small enough that

the interference fringe frequency does not exceed half the range sampling frequency; within that limit, the greater the parallax, the greater the altimetric resolution. A suggestion is that different parallax angles be used for terrain of differing altitude variation [51].

The phase difference with respect to scatterer position in an interferogram is given as [51]:

$$\varphi = \frac{2\pi Q b_\perp r_1}{\lambda z_1 \tan\gamma_1}$$

The saw tooth function expressing the distance between fringes in the interferogram is given as:

$$\Delta R = \frac{\lambda z_1 \tan\gamma_1}{Q b_\perp}$$

$$\gamma_1 = \alpha_1 - \beta$$

where α_1 is the look angle of the system, β is the local terrain slope, and z_1 is the radar altitude [51]. The largest terrain slope measurable without aliasing is when the change in phase is π between pixels. For a slant range sampling distance δR_s, the terrain slope with this variation is [51]:

$$\beta_{\max} = \tan^{-1}\left(\frac{2Qb_\perp \delta R_s}{\lambda z_1}\right) - \alpha_1$$

The planar phase, induced by the geometry, can be removed from the image before unwrapping. In rough terrain, a digital elevation model can be used to subtract geometry-induced phase. The remaining phase is then indicative of the topography [59]. The phase changes due to a flat terrain are subtracted from the interferogram phase differences, leaving the phase differences due to topographic variation [51]

$$\varphi = \frac{2\pi Q b_\perp r_1}{\lambda z_1}\left[\frac{1}{\tan(\alpha_1 - \beta)} - \frac{1}{\tan\alpha_1}\right]$$

The phase unwrapping period of the image (2π) corresponds to a drop in terrain of [51]

$$\Delta z = \frac{\lambda z_1 \sin\alpha_1}{Q b_\perp}$$

The purpose of phase unwrapping is to reconstruct the absolute phase function $\varphi(x,r)$ from the interferometric phase $\psi(x,r)$. The relationship is expressed as $\varphi = \psi + k2\pi$ where $-\pi < \psi \leq \pi$ and k is an unknown integer constant [29, 26].

The phase of the interferogram has a 2π ambiguity, as any phase of x radians is indistinguishable from a phase of $x + 2n\pi$ radians, where n is any integer. An algorithm that is sensitive to phase changes in a two-dimensional field and that tracks the phase over the entire image relative to a given starting position is used. An appropriate multiple of 2π is added at each measurement value, i.e. one that minimizes the phase change between adjacent points. The ambiguous $2n\pi$ phase is thus "unwrapped" at every point. If unwrapping is done on a relative basis, i.e. from point to point in a contour around the interferogram, any error will propagate down the unwrapped phase function. The error will occur where the true phase lies outside the assumed range of less than half a cycle from one point to the next, for example where a steep gradient causes a change in phase greater than half a cycle between image points.

A global ambiguity of $2m\pi$ must be removed by knowing the position of at least one point in the image [68]. Madsen and Zebker [44] describe a phase unwrapping algorithm which does not require ground points to determine the absolute phase of points in the interferometric image.

Phase unwrapping for along track interferometry is also problematic. A natural phase discontinuity occurs due to adjacent areas of ocean being dominated by oppositely travelling Bragg waves. The dominant Bragg waves can change over a distance of as short as a few metres, and this causes the velocity to jump in the ground range plane of $\pm\sqrt{\frac{g}{k_b}}$ [7]. If the interferometric phase jump is greater than π there is confusion in the unwrapping algorithm. The requirements this places on the along track baseline time lag is that the baseline must be short enough so that the phase change due to approaching and receding Bragg waves is less than π :

$$\tau < \frac{1}{4}\sqrt{\frac{\pi\lambda}{g\sin\gamma_1}}$$

A system with multiple baselines can use shorter baselines to aid unwrapping longer ones [7].

If the antennas are not colinear with the radar platform motion, there will almost always be a phase difference between the images. The radar cannot distinguish between sideways motion of the platform and radial motion of the ocean, so points on land are often used to estimate radar attitude. Accurate navigational information is necessary for accurate results in the absence of land for calibration [25].

Residue Methods

In this approach, a two dimensional image is unwrapped along a suitable contour, avoiding discontinuities which would cause phase error propagation [26, 12]. By placing a barrier or 'cut' between points at the extremities of a bad region and unwrapping along contours which go around the cut, error propagation can be avoided. The simplest means of identifying such regions is to make use of the property that the unwrapped phase function satisfies $\oint_c \Delta\varphi \cdot \mathbf{dr} = 0$ around any closed loop, or $\sum_{loop} \Delta\varphi_i = 0$ in its discrete form. The minimum loop is four pixels, or a 'quad'. Along a discontinuity, the loop crosses it twice so the net sum will still be = 0. If the sum of differences around the quad is non-zero, a residue has been detected. These residues are the detected ends of 'ghost lines' across which the phase jumps by one cycle. Residues at opposite ends of a ghost line have opposite signs. Sometimes the ghost lines lie closer together than the sample spacing, resulting in a discontinuity across which the phase jumps by more than one cycle. The objective of residue algorithms is to find the residues for all quads, then link them by suitable cuts. The algorithm determines which residues to pair, and the route of the cuts [29, 26].

Goldstein *et al* suggest straight line joins, chosen to minimize the total cut length [26, 43, 41, 42]. Cuts are formed by pairing off complementary residues and linking them with straight ghost lines. If there exist $2N$ residues, there are N possible partners for each residue and $N!$ combinations of lines. It is not practical to try all combinations, thus less computationally intensive algorithms were devised which do not guarantee the objective of minimal total cut length but are practically realizable [26].

Rocca and Prati [50] investigate a method for locating ghost lines connecting complementary residues based on use of amplitude information in the SAR images. The discontinuities often run along shadow and layover regions of the image. Pairs are connected along lines of minimum cost, where cost rises with line length and variability of backscatter amplitude along the route. In layover areas, energy backscattered by many reflectors falls into a few slant range cells and the amplitude in the image is correspondingly high. Complementary residues falling along these regions will tend to be linked together.

Wilkinson [29] modifies Goldstein's algorithm by combining the idea of residue detection with energy minimization. The phase difference of a pixel pair is estimated by using several pixels in a neighbourhood of the two pixels

instead of just the two pixels concerned. The method reduces the number of residues, eliminating those falling within the neighbourhood taken to estimate the phase difference.

Fringe Detection Methods

Lin, Vesecky and Zebker discuss fringe-line detection by enhancement, thinning and linking algorithms. Since the algorithm uses the original phase difference image and the detected fringe lines, without low-pass filtering, there is no degradation in resolution [40, 43]. Qin and Vesecky propose using edge detection to find the location of fringe contours [41]. The interferogram is filtered to reduce noise. Thereafter edge masks are applied to enhance the fringes, which are then thresholded. The detected fringe points are sequentially linked to form contours, and the phase is unwrapped by adding or subtracting a cycle every time a fringe line is crossed. The method has been tried on SEASAT images of Death Valley, California [41]. The results were improved by using a least-squares fitting of sinusoidal or piecewise-linear basis functions to the edge-enhanced phase image to improve fringe-line detection [42, 43]. Difficulties were initially experienced with a large baseline, where the fringe contours are spaced very closely.

Least Squares Methods

This method involves optimizing a figure of merit for the entire surface. Giglia and Romero [23, 63] propose a figure of merit:

$$F = \sum_{i=0}^{M-2}\sum_{j=0}^{N-1}(\varphi_{i+1,j} - \varphi_{i,j} - \Delta^x_{i,j})^2 + \sum_{i=0}^{M-1}\sum_{j=0}^{N-2}(\varphi_{i,j+1} - \varphi_{i,j} - \Delta^y_{i,j})^2$$

where

$$\Delta^x_{i,j} = W(\psi_{i+1,j} - \psi_{i,j})$$
$$\Delta^y_{i,j} = W(\psi_{i,j+1} - \psi_{i,j})$$

and $W()$ is the wrapping operator which rearranges the phase into the principal phase range.

2.3.5 Phase to Height Conversion

The part of the InSAR processing is carried out using the reconstruction and phase to height conversion equations. A major problem is the availability of accurate estimates of all the system parameters including system geometry, radar frequencies, propagation velocities and unknown phase offset. Not all of these parameters are known and those which are, are known to varying degrees of accuracy. In addition, there may exist more information that is required to solve the problem. A systematic approach to estimating the parameters using all the available information can be devised using the least squares method with an error function that is to be minimized. Wilkinson describes the use of the Chi-square estimation technique for InSAR parameter estimation and error analysis [66].

2.3.6 Image Rectification

The InSAR geometry equations are used to obtain ground range to the target, a process known as rectification. Multipath is a feature displayed often by man-made targets such as bridges. Rectification places the multiple images on top of one another, at different heights. The image with the smallest slant range is then chosen. The ghost images are discarded. Foreshortening, also a phenomenon of slant range images, where mountains have their leading edges compressed and their trailing edges lengthened, is also removed by rectification algorithms [68].

2.4 Error Estimation in InSAR

Error sources in InSAR lead to phase inaccuracies and decorrelation of the interferogram phase, which affects the accuracy of the altimetric resolution in across track InSAR, or the velocity resolution in along track InSAR.

2.4.1 Sources of Error in InSAR Systems

The sources of error can be classified into four sections, deriving from radar limitations, system geometry, terrain and environment, and data processing errors.

Radar Limitations

1. Finite range resolution: the radar bandwidth determines the range resolution for pulses. Each range bin contains the coherent sum of reflections of the scatterers in that resolution cell. The non zero resolution cell size results in speckle. An uncertainty in the height estimation occurs proportional to the displacement of the effective scattering centre [11].

2. Finite azimuth resolution: The focussed resolution in azimuth is limited by the aperture of the real antenna. The along track resolution affect the phase measurement [27].

3. Slant range resolution: Slant range uncertainty refers to the uncertainty as to the distance to the centre of a range bin due to system clock timing, jitters in sampling clock and propagation delay through the atmosphere [39].

4. Receiver Noise (SNR) and Speckle: Receiver noise corrupts phase measurements. There will always be some low amplitude echoes in some resolution cells because of speckle. The receiver noise significantly affects those parts of the image where the reflected energy is low. For narrow baselines where the difference in look angle is small, the measured phase is required to a high degree of accuracy. Thus receiver noise becomes an important factor. Multiple looks, averaging over a number of pixels, are often used to reduce speckle [39]. Receiver noise in a one-pass system can cause interchannel phase errors [11].

5. Transmitter and Receiver Frequency Stability: A coherent reference signal is necessary since angle measurement is a measurement of phase from spatially separated positions. The synthetic aperture length can be limited by the stability of the transmitter or the receiver [32].

6. Quantization noise: The finite resolution of the ADCs and digital storage formats contribute as sources of error [11].

7. Range ambiguities: The sidelobe performance of the system is important because it must be tailored to have low sidelobes to avoid signal energy from neighbouring resolution cells from washing out and corrupting a measurement [11]. Range ambiguities and antenna sidelobes

cause energy from other resolution cells than the one of interest to be picked up, at different phases, so creating interferometric phase errors [32, 3].

8. Azimuth ambiguities: Azimuth ambiguities and sidelobes cause phase errors because the returns do not fit into the interferometric geometry model [11].

System Geometry

1. Baseline decorrelation: this refers to phase errors caused by looking at the projected resolution cell from differing angles. The phase error is a function of wavelength, look angle, range resolution and local terrain slope. Although increasing the baseline length increases the height measurement accuracy, increasing it too far decorrelates the images [39]. Any changes in the baseline length due to flexing of the aircraft airframe change the relative positions of the antenna phase centres and affect the estimate of the interferometric angle. If the difference in look angle exceeds some maximum value, the phase coherence breaks down and the interferogram fringes are no longer visible [11].

2. Baseline length and orientation: The baseline parameters must be known to a very high degree of accuracy for accurate topographic reconstruction. It is one of the most important performance parameters of the interferometer, determining the angular accuracy of the instrument [27]. The position of the radar platform in three dimensions must be known via geopositioning systems (GPS) and other navigational aids to allow the three cylindrical coordinates to be estimated with sufficient accuracy to generate a useful topographic map [69]. Usually the only way to obtain estimates of baseline parameters is through use of calibration points to 'back-calculate' these parameters [45, 40].

3. Vibrations and flexing of the physical baseline: in dual antenna single pass systems and aircraft based systems, physical flexing and vibration of the baseline may contribute as an error source. Careful design is necessary to minimize the effect [11].

4. Motion compensation: roll and pitch in dual antenna systems are compenated for by motion compensation algorithms [11]. Errors in these

algorithms result in errors in the corrections applied. Long term drifts can be removed by calibration points.

5. Calibration point errors: the location accuracy of the calibration points determines the achievable accuracy of the reconstructed surface.

Terrain and Environment

1. Temporal decorrelation: (Repeat pass single antenna systems only) In repeat pass systems, small movements of the target surface between passes on the order of a wavelength constitute a major source of error. Thus the time elapsing between passes must be considered when selecting repeat pass in satellite interferometry.

2. Shadow effect: it is only possible to reconstruct areas of the surface which lie directly in the line of sight of the antennas. Areas not visible to the radar are termed shadowed areas and are characterized by low amplitude returns. This effect occurs for small grazing angles. Satellite systems with small off nadir angles therefore have no problems with shadow. Shadow regions cannot be reconstructed by radar interferometry [11].

3. Layover effects: Areas of the image where the gradient is so steep that two or more distinct sections of the terrain are mapped into the same range cell result in layover. The effect is characterized by bright returns and noisy interferogram phase. A small incidence angle compared with terrain slopes will result in significant layover. Range layover must be solved by processing algorithms, otherwise the slant range/nadir angle ambiguity makes interferometry in such areas impossible [40].

4. Micro-topology: Ideally, interferometric phase should correspond to a point target located at the centre of a resolution cell. Within each cell however are many scatterers, at different heights and locations. The interferometric phase is the combination of contributions within the resolution cell. Microtopology plays a role because each resolution cell contains scatters at different heights and depending on which scatterers dominate, different heights will be estimated from different angles [11].

5. Local slope: the size of the projected range cell depends on the local terrain slope. Steeper inclinations towards the radar increase the base-

line decorrelation and ultimately result in layover. A trade off between shadow and maximum imagable slope is necessary [66].

6. Multiple reflections (Multipath): these essentially result in a form of layover. Where multipath reflection is strong, it may be possible to identify and reconstruct a local picture of the reflected section. However, the absolute location of the area will be displaced in the image. Multipath reflections which are different on each channel or significantly longer than the direct path lead to interferometric phase errors [11].

7. Atmospheric effects: at high microwave frequencies, atmospheric effects such as rain may cause significant phase delays (which may only affect parts of the image). The effects may be difficult to compensate for and further investigation is required to quantify the errors [66].

8. Surface reflectance: in areas of the image where little or no energy is reflected in the direction of the antennas, phase measurements are likely to be corrupted. Very smooth surfaces such as small bodies of water on windless days will have low amplitude returns [66].

Data Processing Errors

1. SAR processing noise: SAR focussing algorithms may introduce additional phase errors owing to quantization effects and other approximations. The front end SAR processor should be high precision, phase preserving, designed to minimize phase noise. The SAR images used to create the interferogram need to be processed at the same doppler centroid frequency to preserve coherence, even though this may be non-optimum in terms of signal to noise and signal to ambiguity ratios [39]. A difference in the signal path length of the two channels creates inter-channel registration and phase errors which have to be corrected using an interpolator and phase calibration in the processing [11]. Inertial navigation system sensing errors affect the solution for terrain height. Finally SAR processing noise due to lack of adequate control over quality parameters such as focussing, sidelobes, dynamic range, arithmetic noise and phase integrity creates errors.

2. Registration errors: Misalignment of the two SAR images results in increased phase noise on the interferogram. Pixel misregistration errors

occur in a similar way, except that now the actual areas from which coherent interference of targets originates are different and therefore results in noisy interferograms [39, 1].

3. Phase unwrapping errors: Any errors in the phase unwrapping process translate directly into distortions in the reconstructed height map. The phase unwrapping process contains potential for global and local errors [11].

2.4.2 InSAR Parameter Sensitivity

Equations for the sensitivity of phase measurement accuracy to InSAR parameters can be derived by differentiating the geometric equations with respect to the InSAR parameters. Sensitivity equations for the standard geometry for across track InSAR used in this thesis are contained in Chapter 4.

There are two types of errors which affect the accuracy of InSAR results. The first type is a posting error, where the estimated measurement derived from the unwrapped interferometric phase is wrongly located on the map. The second type is an error in accuracy, where the estimated measurement derived from the unwrapped interferometric phase is incorrect. Both types of error can occur as local errors, in which case they occur in isolated pixels, or as global errors, where the error is systematic. Systematic errors can always be removed through the use of calibration points.

Motion Compensation

Motion compensation is critical for accurate topographic maps [45, 28]. The pixel dependent phase correction required depends on the elevation of each scene patch, which is initially unknown, so requiring a flat earth assumption. This leads to terrain induced phase errors in the compressed data. Aircraft roll-error is directly proportional to cross-track distance and antenna look-angle [11, 69]. The effects of roll-error are most apparent in the far-range part of the swath [68].

Phase Decorrelation

Various researchers have tackled the subject of quantifying these effects [39, 55, 67, 1]. Phase decorrelation is due to thermal noise, speckle, pixel misregistration and baseline decorrelation. Thermal noise is dependent on

the signal to noise ratio, while speckle noise depends on the number of coherent looks taken. Receiver thermal noise increases with range, both because of the interferometric equations and because the SNR decreases with range. The larger the baseline, the less the effects of additive thermal noise [52]. Baseline speckle is affected mainly by the incidence angle, because of the change in slant range resolution [11]. It is almost independent of baseline length [52].

Pixel misregistration is not be discussed in this thesis.

The spatial baseline, rotation and temporal correlation functions depends on parameters of either the sensor or the target [67, 1]. For repeat pass interferometers, the changes occurring on the ground describe processes occurring on the timescale of orbit repeat times and size scales of the radar wavelength. The orbit selection is therefore driven by the combination of error level sources, such as attainable baseline and expected time decorrelation from the regions to be mapped. For single pass interferometers, only the first two terms apply, and they drive the selection of baseline length, radar wavelength and look angle for a given type of terrain [55].

Phase noise can be quantified by the phase variance, which is related to the *correlation coefficient* γ between two received signals $\mathbf{v}_1$ and $\mathbf{v}_2$. The correlation coefficient is defined as [55]:

$$\gamma = \frac{|\langle \mathbf{v}_1^* \mathbf{v}_2 \rangle|}{\sqrt{\langle |\mathbf{v}_1|^2 \rangle \langle |\mathbf{v}_2|^2 \rangle}} = \frac{|\alpha|}{1 + \frac{1}{SNR}}$$

where the parameter α is the geometric correlation in the absence of receiver noise [55].

The correlation parameter is often expressed as the *decorrelation coefficient* $\delta = 1 - \gamma$. The relationship between phase variance σ_ϕ and the decorrelation coefficient has been evaluated by simulation and results presented in the form of graphs in Chapter 4 [39, 55]. The dominant spatial decorrelation source is the baseline noise caused by viewing the surface at slightly different aspect angles. Effects relating to changes in the target surface with time are known as temporal effects. This causes decorrelation which is related only to the properties of the surface.

The total decorrelation δ_{total} can be written as

$$\delta_{total} = \delta_{temporal} \cdot \delta_{spatial} \cdot \delta_{thermal}$$

Thermal correlation in the event of equal thermal noise powers at each antenna is given as $\frac{1}{1+SNR^{-1}}$

The spatial baseline decorrelation [67] increases with increasing b, because this increases the difference in look angles between the two antennas. The critical baseline length, at which backscatter at each pixel is spatially completely uncorrelated, is [69]

$$b_{c\perp} = \frac{\lambda r_1}{Q\delta R_s} \tan \alpha_1$$

At the same time, sensitivity of the InSAR phase to height changes increases with an increasing baseline. Thus the baseline must be long enough that differences in phase between resolution cells are detectable above the noise threshold, but short enough that the spatial decorrelation does not degrade the signal. The decorrelation can be reduced using spectral filtering during range compression in the SAR data correlator, for example the ω-k algorithm [70]. In Chapter 4 a method of obtaining an optimum baseline is described. Implementations of interferometers generally use a baseline length of between 0.2 and 0.8 of $b_{c\perp}$ [55]. Errors in the knowledge of baseline elevation angle translate to errors in local surface slope estimate. This systematic error can be corrected using tie points.

Crossed orbit interferometry baseline constraints are more severe than for parallel orbit interferometry [19]. Increased height estimation errors occur as the incidence angle decreases [11]. Tunable InSAR has been proposed to remove the ground spectrum frequency shift caused by the baseline [21]. Alternatively, only the frequency band common to both images can be kept [12, 1].

The sensitivity of SAR interferometry therefore depends on the observation geometry. For non-zero baselines the sensitivity is a complicated function of image signal to noise ratio and baseline length [39]. Double-difference interferograms are even more complicated as they result from two single-difference images. Averaging over pixels can increase the sensitivity at the expense of spatial resolution.

Spatial decorrelation due to rotation of the target with respect to the radar look direction that takes place between the illuminations of the target by the antennas is negligible for near-repeat orbits and does not apply for single pass systems [67].

Temporal decorrelation is caused by terrain movement. The contribution to height error is unaffected by range, but instead by beam incidence angle [11]. It is significant for incidence angles of less than 45° in that the sensitivity is greater for vertical changes than horizontal changes, so forests and other

surfaces with significant volume scattering decorrelate most rapidly. For L-Band, 10 cm of rms motion is enough to decorrelate the signal completely, while for C-Band it is about 2-3 cm rms motion [67]. An advantage of single pass systems is that they avoid the problem of temporal decorrelation [69]. For satellite systems, single pass configurations do not yet exist, and this limits the usefulness of satellite systems to terrain which does not decorrelate much over time. However, satellite systems can cover terrain inaccessible by airborne systems [69].

The decorrelation length of the baseline is defined by the scene coherence time and this can cover a wide range of values within a single scene. For across track interferometry the decorrelation length of the baseline is a geometric process to first order, and is fixed for a given scene, changing little within it. The finite signal to noise ratios associated with the data from each antenna will affect the correlation observed [6, 7]. For along track InSAR over the ocean, the decorrelation length or time between images is much shorter, and may vary widely throughout the swath. A dual-baseline method of obtaining the coherence time of the ocean surface, using the C and L bands of the AIRSAR platform exists [6, 7]. Using this method, two independent measurements of interferometric velocity are made, two of the four resulting baselines are combined to increase the signal to noise ratio, and the other two measurements combined yield a more accurate estimate of the coherence time.

The total decorrelation δ_{total} is the multiplication of the three derived decorrelation functions. The phase standard deviation can then be calculated and the height error associated with it [67]. The phase variance can be reduced by multi-looking [51, 55]. The Cramer-Rao bound for the phase standard deviation has been shown to be

$$\sqrt{\sigma_\varphi} = \sqrt{\left\langle \hat{\psi} - \left\langle \hat{\psi} \right\rangle^2 \right\rangle} = \frac{1}{\sqrt{2N_l}} \frac{\sqrt{1-\gamma^2}}{\gamma}$$

where N_l is the number of independent looks. This simulation is carried out in Chapter 4.

2.4.3 Ambiguity Analysis

The interferometric phase is measured modulo 2π. As the baseline is much longer than the wavelength, the reflection corresponding to a single phase

measurement might have come from one of several directions and is therefore ambiguous. Over a continuous terrain, the corresponding interferogram appears as a series of fringes. In this section we derive formulae for calculating the fringe spacing on an interferogram, angular ambiguities and spatial ambiguity spacings. Phase unwrapping is difficult for small altitudes of ambiguity [66, 50, 47].

Interferogram Fringe Spacing

The interferogram fringe spacing refers to the spatial separation over which the interferometric phase changes by one cycle. For a flat, horizontal surface, the interferogram appears typically as a series of parallel, approximately uniformly spaced fringe contours. The fringe spacing decreases gradually as a function of range across the interferogram.

It is important to know the fringe spacing since it is an essential parameter in the design process of an InSAR system. If the interferogram is to be successfully unwrapped, then the phase should be spatially sampled at least twice in every cycle, in practice at five to ten times per cycle for more noisy interferograms [51].

In the configuration depicted in Figure 2.14 showing a flat surface, inclined at an angle γ_1 with respect to the incident ray, the phase ambiguity is one cycle: $\Delta\varphi_{amb} = 2\pi$ For the case of $r_1 \gg b$, the following approximations can be used:

- angular ambiguity spacing:

$$\Delta\theta_{amb} \simeq \left| \frac{\lambda}{Qb\cos\theta_1} \right|$$

  interferogram fringe spacing:

$$\Delta r_{amb} \simeq \left| \frac{\lambda r_1 \tan\gamma_1}{Qb\cos\theta_1} \right|$$

Effective Location Ambiguities In the phase unwrapping process the absolute phase value φ is computed from $\varphi = \psi + k2\pi$, where k is an integer. Any error in k will result in the reconstructed point being in error. If the unwrapped value is in error by N cycles, then the location errors will be $N\Delta l_{amb}$, $N\Delta y_{amb}$ and $N\Delta z_{amb}$, where Δl_{amb} is the shortest distance between

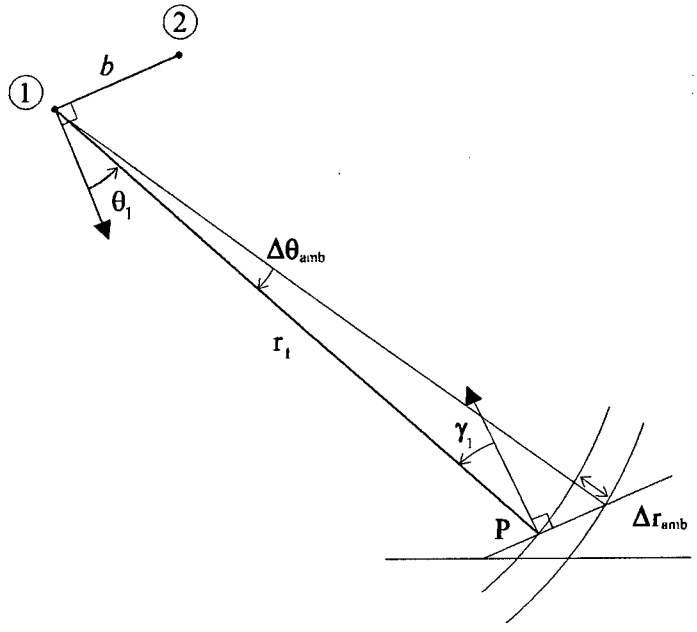


Figure 2.14: Geometry for Fringe Spacing Calculations

two adjacent ambiguities, Δz_{amb} is the vertical displacement between two adjacent ambiguities, Δy_{amb} is the horizontal distance between two adjacent ambiguities.

Ambiguity Separations for Flat Earth Geometry Figure 2.15 shows the geometry of ambiguity spacings [66]. The approximations are accurate if $r_1 \gg b$:

$$\Delta l_{amb} = \Delta\theta_{amb} r_1$$
$$\Delta z_{amb} = \Delta\theta_{amb} r_1 \sin\alpha_1$$
$$\Delta y_{amb} = \Delta\theta_{amb} r_1 \cos\alpha_1$$

Ambiguity Separations for Spherical Earth Geometry The approximations are accurate if $r_1 \gg b$. The equations are identical, with α_1 replaced by η_1, where:

$$\eta_1 = \alpha_1 + \phi_1 = \alpha_1 + \tan^{-1}\left(\frac{r_1 \sin\alpha_1}{A_1 - r_1 \cos\alpha_1}\right)$$

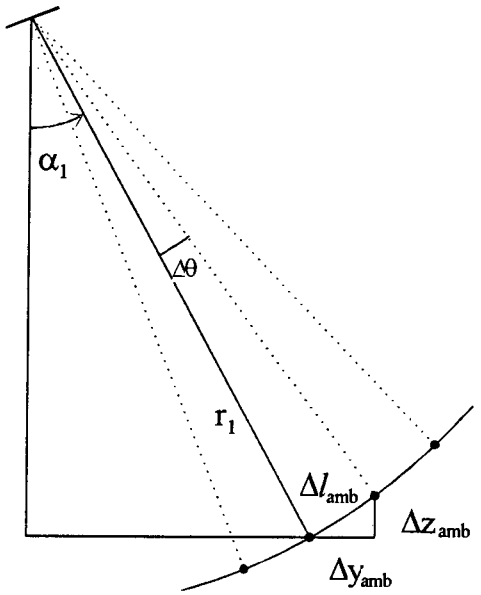


Figure 2.15: Ambiguity Spacing for Flat Earth Geometry

2.4.4 Layover and Shadowing Constraints

The maximum and minimum slopes that can be reconstructed are defined with the aid of Figure 2.16 [66]. Layover occurs for all incident angles $\gamma_1 \leq 0°$. Shadowing occurs for incident angles $\gamma_1 \geq 90°$. The range over which imaging is possible is

$$0° < \gamma_1 < 90°$$

The surface slope β is defined as the angle between the surface normal and the vertical, being the normal to a defined reference horizontal plane

$$\beta = \gamma_{hor} - \gamma_1$$

Shadowing occurs for slopes $\beta \geq \gamma_{hor}$. Layover occurs for slopes $\beta \geq \gamma_{hor} - 90°$. The range of slopes over which topographic reconstruction is possible is

$$\gamma_{hor} - 90° < \beta < \gamma_{hor}$$

It is not possible to reconstruct the topography of areas where the surface slope is not within this range, except through the use of additional information of interpolation. The steepest possible slopes prescribed by layover and shadowing, in terms of the notation used, are [66]:

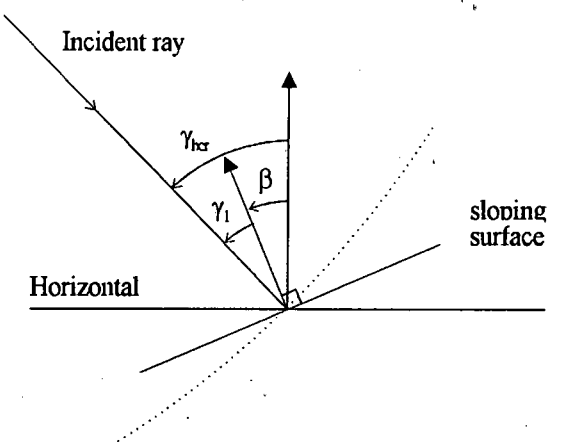


Figure 2.16: Geometry defining Surface Slope

- for a flat earth geometry

$$\gamma_{hor} = \alpha_1$$

- for a spherical earth geometry

$$\gamma_{hor} = \eta_1$$

Chapter 3

Review of InSAR Implementations

Radar interferometry is a technique for topographic mapping and feature recognition. High resolution digital elevation maps can be achieved using airborne or satellite-based InSAR. The radar platform can perform InSAR with either one or two antennas mounted, different procedures being followed in these respective cases.

InSAR is capable of generating topographic maps of the terrain in across track configuration or of measuring the across track surface velocity of the ocean waves in along track configuration.

The technology required to implement InSAR is well-defined. A radar platform with one or two antennas is required, together with a phase preserving radar signal processor. Since radar interferometry was first proposed in 1974, a variety of processing methods and algorithms have been developed. More efficient algorithms are under investigation in the USA, Canada and Europe [70, 23, 29, 21, 1, 60].

InSAR is today a feasible technology. A mark of its perceived relevance and importance is the ongoing research being done to improve the processing algorithms, to increase their reliability, speed, flexibility and ability to deal with the variety of problems induced by different operational requirements.

3.1 Historical Development of InSAR

Radar interferometry as a method of mapping the Earth's surface from aircraft was first proposed in 1974 by Leroy C. Graham [27] of the Goodyear Aerospace Corporation. The research was undertaken under joint sponsorship of the U.S. Air Force Wright Air Development Center and the U.S. Army Engineer Topographic Laboratories. The system designed was put into service with the Air Force Charting and Geodetic Squadron of the Air Weather Services under supervision of the Defense Mapping Agency. It comprised two antennas mounted on an aircraft, displaced across track facing at right angles to the aircraft flight path.

Previously to this development, radar interferometry had been used in astronomical research. In 1969 interferometry was used to separate ambiguities from the northern and southern hemispheres in earth-based observations of Venus. The first altimetric measurements were made in 1972, when the surface of the moon was mapped. In 1974 the same was done for Venus [68].

In 1986, after the advent of capable array processors, Howard A. Zebker and Richard M. Goldstein from JPL/NASA developed the CV-990 airborne InSAR system. The system was used to image San Francisco Bay in the along track configuration with great success [68, 67].

Since 1986 InSAR studies have been carried out in various countries around the world. Among them are Canada, where the CCRS is operating the C/X-SAR system, Germany, where Dornier and Deutsche Aerospace have since 1994 a working system, Japan and Italy [51]. A variety of satellite SAR systems launched since 1994 includes SIR-C. This study limits itself to documenting systems operating before May 1994 [18].

3.2 Theoretical Implementations of Radar Interferometers

Two distinct implementation approaches can be taken which differ in how the interferometric baseline is formed. These are single pass and repeat pass (also known as repeat pass) systems:

- Single pass (dual antenna) Interferometer

  Two antennas are mounted on the radar platform, displaced either across track to measure height, or along track to measure velocity.

Most airborne systems operate in this manner.

The across track system is used for topographic applications, while the along track system is used for measuring ocean currents.

- Repeat pass (single antenna) Interferometer

  One antenna is mounted on the radar platform, which has to overfly the target area twice to obtain the images required for interferometry.

  Most satellite interferometry is done in this way, images being made on successive orbits.

3.2.1 Repeat Pass Interferometers

In the repeat pass interferometer a single antenna and radar system illuminate a surface at different times and at slightly differing geometries, to create a synthetic interferometer. A significant amount of data processing is required in terms of motion compensation. A problem also to be considered when using repeat pass interferometry for terrain elevation mapping is the possible loss of coherence between passes [28]. This implies that there should be no significant change in the radar backscatter of the terrain between passes. Changes in the surface, on the wavelength scale, lead to decorrelation noise in the interferogram. If the change owing to, for example, precipitation, is large enough, it could result in complete temporal decorrelation of the received signals. Apart from the problem of temporal coherence loss, equations for the difference in slant range to the observed point from the antenna on its different passes can be derived, and from this a phase difference. The look angle to the observed points can be determined from this, and height values obtained [28].

An extension of the repeat pass radar interferometry is differential interferometry. Here two interferograms are created from three (or more) SAR images of the terrain, taken at different times on repeated passes on slightly different tracks. The baseline lengths are again subject to the maximum length restriction that the speckle in the image must not change substantially, i.e. both observations for an interferogram need to be within the effective beamwidth of the reradiating resolution element. The two interferograms now have their phases unwrapped, to create topographic maps in conjunction with the intensity information. This is necessary as the phase differences cannot be simply subtracted. Next the phases of the two maps are

plotted against each other, each scaled by the length of the baseline used to obtain them. Deviations from an ideal ellipse tilted at 45° give information about surface changes. The two maps' scaled phases are subtracted from one another to create a double-difference interferogram. The phases due to the topography are hereby removed, and what remains is a phase image showing the changes in the topography between the times of the observations [20]. Since the changes all occur at the short carrier wavelengths, the sensitivity of the system is limited only by the phase noise of the system, i.e. the statistics of the speckle. It is noteworthy that if there were a zero baseline, only two observations would be necessary to obtain differential interferometry. The non-zero baseline causes the topography information to be mixed with the surface changes, so a third observation becomes necessary.

Only one-dimensional changes along the line of sight are measured, so recovering entire surface motions requires other sets of interferograms made along other directions, and resolving the observations into their vector components.

In correlating raw images prior to interferometric processing, the azimuth focussing for all three needs to be the same, otherwise there will be errors in the phases, with associated phase coherence loss.

A vector coordinate system using normal, tangential and across track vectors can be used to estimate the baselines of ERS-1 satellite orbits instead of using only the minimum distance at one point. The vectors can be used to generate coarse offsets for image registration. Quadratic interpolation on sections of the image with coarse shifts added in is used to maximize the interference fringe visibility function [59].

3.2.2 Single Pass Interferometers

Single pass interferometry involves forming a baseline by means of two physical antennas mounted on the radar platform some distance apart. These antennas illuminate the same area of ground at the same time. One of the antennas is used to transmit, while both are used to receive. The single pass method avoids problems with temporal decorrelation of the surface. This approach has been used on aircraft [27, 68] and an implementation in space using tethered satellites to create an adequate baseline has been proposed [49].

3.2.3 Tunable Interferometric SAR

Tunable interferometric SAR systems have been proposed [21]. This proposal alleviates the problem of spatial baseline decorrelation. Whenever a change in look angle occurs, such as in an interferometric system with a baseline, the frequency spectra for the ground returns for each of the antennas belong to different bands of the ground reflectivity's spectrum. The same spectral components as in the first signal are found in the second, shifted by a frequency dependent on the difference in look angles between the antennas. This effect is the same as that observed in optical diffraction gratings. When the spectral shift equals the system bandwidth, the critical baseline is reached. The equation for critical baseline derived using spectral shift is equivalent to the one derived using geometric decorrelation. A repeat pass Tunable InSAR system would shift the transmitted signal by a frequency equivalent to the shift introduced by the baseline length. Thus longer baselines could be used. On single pass InSAR systems, the transmitter bandwidth would have to be greater than the receiver bandwidth so that a bandwidth centred around two separated centre frequencies could be transmitted when appropriate.

3.3 InSAR Systems

Several countries have investigated or are investigating InSAR technology. Some have operating InSAR systems, others have InSAR processors only, while still others research particular topics in the InSAR field.

3.3.1 InSAR Platforms

In May 1994 there were three operating airborne systems. These are located in the USA, Canada and Europe.

- The systems built by NASA/JPL in the USA
- The system built by the UK Defence Research Agency in the United Kingdom
- The system built by the CCRS in Canada

NASA/JPL InSAR Systems

Two systems have been operated by NASA/JPL. They are the CV-990 system, and the AIRSAR/TOPSAR system.

CV-990 System The system was mounted on a CV-990 aircraft. Initially two L-band antennas were fitted in the along track configuration, along the fuselage, and it was used to obtain velocity of ocean currents [25].

Later, an across track antenna configuration was tried, the purpose being to investigate the accuracy of InSAR in comparison with USGS maps, and then to improve the motion compensation algorithms used. Two L-Band antennas are used, and no GPS was fitted. The two antennas were mounted one on the undersurface of the starboard wing, and one on the undersurface of the port wing [68]. The parameters of the single pass across track CV-990 system are shown in Table 3.1.Results show a height measurement statistical

Table 3.1: CV-990 System Parameters

Parameter	**Units**	**CV-990**
Operating Frequency	GHz	1.2
Wavelength	cm	24.5
Peak Power	kW	5
PRF	Hz	800
Operating Altitude	km	7.8
Velocity	m/s	238
Depression Angle	degrees	20-57
Slant Range	km	20
Swath Width	km	10
Range Resolution	m	11
Azimuth Resolution	m	12
Baseline Length	m	11.1
Baseline Elevation	degrees	0

variation of 2 - 10 m, and a high degree of correlation with USGS maps.

The CV-990 system was used to generate 10 m resolution SAR images which were then combined using across track interferometry [68]. The geometrical relationship of InSAR allows determination of the height of ground

points. The height, slant range and azimuth distance parameters allowed rectification of SAR images.

The CV-990 system in its along track configuration was used to obtain ocean images in the San Francisco Bay area [25]. The velocity at each pixel is the sum of the line of sight components of the current, swell and Bragg waves.

Unfortunately the CV-990 and JPL radar system were destroyed in a fire in 1985, and JPL subsequently developed the AIRSAR system [6].

AIRSAR/TOPSAR System The TOPSAR (Topographic SAR) system was a prototype C-Band InSAR system. It was an addition to the existing JPL AIRSAR (Airborne SAR) multi frequency multipolarimetric system, mounted aboard a NASA DC-8 aircraft. The original AIRSAR system mounted P (68 cm), L (24 cm), and C-Band (5.6 cm) radars. After the destruction of the CV-990 system, JPL used a similar along track L-Band configuration on the DC-8 aircraft which could generate two interferograms in two baselines [6]. The P-Band radar was not used for InSAR. Table 3.2 contains the AIRSAR specifications.

The AIRSAR system was used to measure along track ocean currents and shoaling in 1989 in the Monterey Bay area [46] and in 1994 in the Messina Straits area [7]. The coherence time of the ocean was also measured [7]. Even water appearing smooth in the phase image shows structure in the coherence image. Upwelling and features observed in phase and amplitude images are clearly defined as lines of longer coherence times.

Carande [6, 7] describes a dual-baseline method of obtaining the coherence time of the ocean surface, using the C and L bands. Using this method, two independent measurements of interferometric velocity were made, the SNR was increased, and a more accurate estimate of the coherence time was obtained. JPL then optimized the system for topographic mapping by implementing an InSAR system for single pass topographic mapping applications [15]. Two extra antennas were added, displaced across track and operating at C-Band. They are mounted on the fuselage, one underneath the existing P-Band antenna fairing, the other at window level on the port side of the aircraft. The baseline is thus at an angle of 62 degrees with respect to the horizontal. An INS, barometric altimeter, GPS are fitted. The C-Band interferometer replaces one channel of the C-Band polarimeter instrument when in use, but leaves the P and L-Band radars undisturbed [69]. Thus the com-

Table 3.2: AIRSAR System Parameters

Parameter	Units	AIRSAR
Operating Frequency	GHz	5.3 and 1.2
Wavelength	cm	5.6 and 24
Range Bandwidth	MHz	20 or 40
Peak Power	kW	1 and 6
PRF	Hz	600
Operating Altitude	km	9
Velocity	m/s	210
Depression Angle	degrees	30-55
Slant Range	km	10-15
Swath Width	km	12 by 6
Range Resolution	m	6
Azimuth Resolution	m	12
Baseline Length	m	2.58
Baseline Elevation	degrees	62.8

bined system produces fully polarimetric L and P-Band, plus C-Band VV polarized backscatter images in addition to the topographic product when in use. The parameters of the TOPSAR system are given in Table 3.3 [69, 45]. The TOPSAR system was first flown in 1991, imaging swaths of 30 km long by 6.4 km wide. Results showed that over flat terrain, statistical height errors were about 3 to 4 m, while over rugged terrain, height accuracy errors were in the 20 to 40 m range. Systematic height errors of 10 to 20 m were also incurred [69]. These results were achieved without the benefit of motion compensation algorithms. The short baseline meant that errors were not due to spatial decorrelation, but only to finite SNR. When motion compensation was used, the height error was reduced, but dominated by the attitude error, a 0.05° causing a 6.2 m height error.

Research was the initiated into more efficient motion compensation algorithms. These were used in 1993 to process the data and compare results with USGS maps. The errors were reduced to 2 m over flat terrain, and 5 m over rugged terrain [45].

The TOPSAR radar was used in a study of alluvial fans in Death Valley, California [15]. Topographic information replaced manual counting of contour cemulations to obtain an estimate of the age of the fans.

Table 3.3: TOPSAR System Parameters

Parameter	Units	TOPSAR
Operating Frequency	GHz	5.288
Wavelength	cm	5.66
Range Bandwidth	MHz	40
Peak Power	kW	1
PRF	Hz	600
Operating Altitude	km	9
Velocity	m/s	200-300
Depression Angle	degrees	30-55
Slant Range	km	10-15
Swath Width	km	6.6
Range Resolution	m	10
Azimuth Resolution	m	10
Baseline Length	m	2.58
Baseline Elevation	degrees	62.8

TOPSAR was used to investigate lava flows on Mt. Hekla, Iceland, and lava flow fields in Hawaii [15].

UK Defense Research Agency System

This system was developed under contract of the British Ministry of Defense by DRA Malvern and completed in Autumn 1993. The purpose of the system is to investigate the feasibility of short-range SAR, and research MTI in the presence of clutter using interferometric and displaced phase centre antenna methods, ECCM, and military target recognition and classification by polarimetric signature techniques.

The antennas operate at C-Band. The system is installed in a RAF Fokker Andover aircraft based at DRA Farnborough. The three antennas are mounted horizontally aft of the wing on the port side of the aircraft, with the centre antenna fixed, the outer two being movable to widen the distance between them. To keep costs down, no IMU is fitted, but a GPS and INS is fitted [2].

The system has been successfully statically tested, but no official results from a flight are yet available. The parameters of the British system are

shown in Table 3.4.

Table 3.4: British short range SAR System Parameters

Parameter	Units	British
Operating Frequency	GHz	5.7
Wavelength	cm	5.26
Peak Power	W	7
PRF	kHz	10
Operating Altitude	km	0.1 - 3
Velocity	m/s	100
Depression Angle	degrees	0-75
Slant Range	km	10
Swath Width	km	5
Range Resolution	m	2.1 or 21
Azimuth Resolution	m	0.2 or 2
Baseline Length	m	1
Baseline Elevation	degrees	0

The CCRS C/X-SAR System

The CCRS conducted an experiment in 1990 in which repeat pass interferometry was investigated. A Convair 580 aircraft was fitted with an X-Band and C-Band antenna and flown over a test site in Ottawa, Ontario. An INS system was fitted. The results showed that it was possible to obtain interferometric fringes [28, 11]. Since then, two antennas have been fitted. The parameters of the CCRS system are given in Table 3.5.

3.3.2 InSAR Processors

Various countries utilize sets of radar images obtained elsewhere in order to perform interferometric studies with their own processors without actually possessing an operating InSAR system.

The ESA operates the ERS-1 satellite, the Japanese also operate a satellite known as the ERS-1, while Canada operates RADARSAT, and NASA operates SIR-C. SIR-B is a space shuttle imaging radar which obtains radar stereo data. ERS-1, RADARSAT and SEASAT satellite images have been

Table 3.5: C/X-SAR System Parameters

Parameter	Units	C/X-SAR
Operating Frequency	GHz	5.3
Wavelength	cm	5.66
Look Angle	degrees	64.6
Operating Altitude	km	6
Slant Range Resolution	m	5
Slant Range	km	14
Baseline Length	m	2.8
Baseline Elevation	degrees	45

used by NASA, the ESA and various other bodies to research radar interferometry applications [18].

ERS-1 (ESA) This satellite was launched July 1991, into polar orbit of altitude 790 km. The parameters of the system are given in Table 3.6. ERS-1

Table 3.6: ERS-1 System Parameters

Parameter	Units	ERS-1
Operating Frequency	GHz	5.3
Wavelength	cm	5.66
Range Bandwidth	MHz	15.55
Peak Power	kW	4.8
PRF	Hz	1679
Off Nadir Angle	degrees	23
Swath Width	km	100
Slant Range Resolution	m	9.375
Ground Range Resolution	m	25
Azimuth Resolution	m	7
Critical Baseline Length	m	1100
for terrain slope	degrees	0

was operated in close repeat pass orbits during 2 three month phases by 1993. The repeats of the orbital tracks were made every 3 days. Routine image products produced by processing centres contracted to the ESA include vari-

ous radar intensity images and coherent images retaining phase information. Pairs of coherent images are the starting point for interferometric processing.

The baseline $b_{c\perp}$ for flat terrain is 1115 m. With a SNR or 11.7 dB, the thermal decorrelation coefficient $\delta_{thermal} = \frac{1}{1+SNR^{-1}} = 0.94$ [70]. Typically $b_\perp = 165$ m which means that the baseline decorrelation $\delta_{baseline} = 0.85$. The combined decorrelation is then 0.81. Using 10 looks, the rms phase error is 11° which translates into an rms height error $\sigma_h = 1.8$ m. This assumed no data processing or temporal decorrelation errors.

The baseline uncertainty of 30 cm leads, for the given $b_\perp$, to a systematic height error $\sigma_h = 1$ km which can be removed by tie points.

ERS-1 has been successful over lava flow areas, even with a 35 day period between passes. This proved that repeat pass interferometry was possible over selected terrain [70]. However, ERS-1 is not deemed suitable for global mapping because of its small incidence angle, which limits the slopes that can be imaged, and the fact that temporal decorrelation at C-Band is too quick to make repeat pass interferometry viable over all types of terrain.

In 1992 ERS-1 InSAR data was collected over Bonn, Germany, in 3 day repeat pass mode. The Milan Polytechnic group [49] processed the data to create DEMs with vertical 20 m accuracy. Tie points have been used to resolve ambiguities resulting from layover and phase aliasing on slopes [12]. An extension of the tie points principle has been used to do baseline modelling using least squares fit [59]. The solution gives accuracies σ_h of from 59 m for a rough solution, down to 8.2 m for a four parameter non linear least squares fit.

SEASAT SEASAT was launched in June 1978 and was extensively used for interferometric experimentation before ERS-1 appeared. It was the first spaceborne SAR system launched by NASA. It operated for 105 days, collecting 50 hours of SAR data [65]. It operated at an altitude: 800 km with a three day repeat orbit cycle. The specifications are given in Table 3.7. The terrain slopes measurable with SEASAT images is limited by the off nadir angle. For baselines perpendicular to the line of sight of 99 m and 1030 m, the maximum slopes were 22.8° and 21° respectively. For greater slopes, loss of coherence occurs. Phase aliasing occurred at slopes of 22.2° and 15.2° respectively as predicted. The altitude resolution for SNR of 22.7 dB for the short baseline and 43 dB for the long baseline was 30 m and 2 m respectively [51]. SEASAT images of Death Valley, California, been successfully

Table 3.7: SEASAT System Parameters

Parameter	Units	SEASAT
Operating Frequency	GHz	1.275
Wavelength	cm	23.5
Range Bandwidth	MHz	19
Peak Power	kW	1
Off Nadir Angle	degrees	23
Swath Width	km	15 by 100
Slant Range Resolution	m	6.27 or 7.89
Ground Range Resolution	m	20
Azimuth Resolution	m	3.93 or 22.4

processed using InSAR, with a repeat pass time of 3 days [39]. Differential radar interferometry has been investigated using SEASAT data [20].

SIR-B The SIR-B shuttle imaging radar was launched in October 1984 [19]. It was placed in a low orbit near 236 km altitude. Its specifications are given in Table 3.8. The crossed orbit configuration makes SIR-B processing more

Table 3.8: SIR-B System Parameters

Parameter	Units	SIR-B
Operating Frequency	GHz	1.2
Wavelength	cm	23.4
Range Bandwidth	MHz	12
Orbit Angle (2α)	degrees	1.208
Slant Range	km	280
Look Angle	degrees	31
Swath Width	km	100
Ground Range Resolution	m	30
Azimuth Resolution	m	25

complex than parallel orbit systems. Algorithms for crossed orbit InSAR have been tried successfully using SIR-B data [19]. A squint angle was chosen for the SAR Doppler processor to give the best interferogram fringes. One of the two images was resampled in order to register them properly, and

the intensity spectrum registration method was used. Geometrical InSAR relationships for crossed orbits were developed to obtain height information.

Chapter 4

Analysis of the SASAR VHF SAR System for InSAR Operation

SASAR is a locally designed VHF frequency SAR system which has been under development since 1994 [14, 36, 33, 35, 34]. The platform is to be a Boeing 707, and the single VHF antenna is to be attached in a cheek blister on the forward left fuselage [34]. The specifications of the proposed system are shown in Table 4.1. The swath actually used for processing differs from the natural swath mapped by the radar beam. A maximum of 4096 pixels

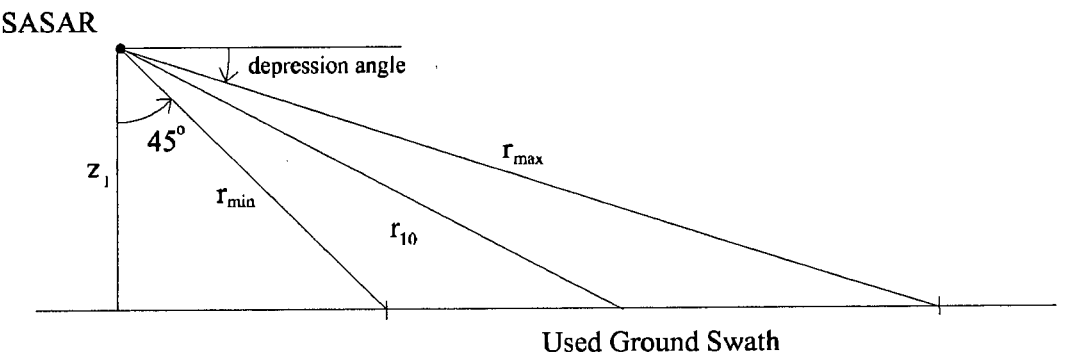


Figure 4.1: Ground Range Swath Geometry of the SASAR System

can be imaged, with range bin size equal to 13 m. This results a slant range

Table 4.1: SASAR System Specifications

Parameter	Units	SASAR
Operating Frequency	MHz	141
Wavelength	cm	213
Pulse width	ns	175
Slant Range resolution	m	26
Azimuth resolution	m	26
Peak Power	kW	1
Receiver Noise Factor	dB	8
Operating Altitude	km	2-10.7
Maximum velocity	m/s	246
Possible PRF range	Hz	177-2000
Nominal PRF	Hz	625
Depression Angle	degrees	16-45
Elevation Beamwidth	degrees	60
Swath Width	km	56.8

swath of 53248 m. The processed swath begins at an angle of 45° as shown in Figure 4.1.

At an operating altitude of 10700 m, the swath is constrained by the antenna elevation beamwidth to end at 10° but the processed swath continues to 9.5° depression angle, resulting in a processed ground range swath of 56643 m. The depression angle at of the centre ground range swath is 14.2°.

At the lowest operating altitude of 2000 m, the swath is constrained by the antenna elevation beamwidth to end at 10° but the processed swath continues to 2° depression angle, resulting in a processed ground range swath of 54040 m. The depression angle at the centre of the ground range swath is 3.9°.

In this Chapter the viability of the SASAR system for InSAR operation is investigated, and a parameter sensitivity analysis is undertaken.

Appendix E contains all the MathCAD worksheets used in the analysis.

4.1 System Analysis

There are five steps to be taken in an analysis of the SASAR system.

1. The geometrical equations describing across track InSAR are differentiated with respect to the InSAR parameters, and the meaning of an uncertainty in a parameter and its effect on InSAR measurements is explained.

2. From the parameter sensitivity equations an optimum baseline elevation angle for a particular antenna depression angle can be chosen.

3. The factors influencing phase estimation error in InSAR are discussed, and an optimum baseline length is found.

4. The sensitivity of InSAR measurements to variations in the system parameters is quantified using the optimum system determined in the previous steps.

5. The suitability of InSAR to map different terrain slopes is examined and the criteria for flat terrain and for slopes are examined.

4.2 Parameter Sensitivity Equations

The equations describing the geometrical radar-target relationship for across track InSAR (see Subsection 2.2.1) are (Q = 2):

$$\varphi = \frac{2\pi Q b}{\lambda} \sin\theta_1$$

$$z = z_1 - r_1 \cos(\theta_1 + \theta_b)$$

Differentiating with respect to the parameters gives the height error due to an uncertainty in that parameter:

$$\sigma_z(r_1) = \cos(\theta_1 + \theta_b)\sigma_{r_1}$$

$$\sigma_z(z_1) = \sigma_{z_1}$$

$$\sigma_z(b) = \frac{r_1}{b} \frac{\tan\theta_1}{\sin(\theta_1 + \theta_b)} \sigma_b$$

$$\sigma_z(\theta_b) = r_1 \sin(\theta_1 + \theta_b)\sigma_{\theta_b}$$

$$\sigma_z(\lambda) = \frac{r_1}{b} \frac{\varphi}{Q2\pi} \frac{\sin(\theta_1 + \theta_b)}{\cos\theta_1} \sigma_\lambda$$

$$\sigma_z(\varphi) = \frac{r_1}{b} \frac{\lambda}{Q2\pi} \frac{\sin(\theta_1 + \theta_b)}{\cos\theta_1} \sigma_\varphi$$

4.2.1 Parameter Uncertainty Definition

The height errors σ_z are due to uncertainties in the InSAR parameters.

Slant Range

The uncertainty σ_{r_1} in slant range measurement refers to the error in determining the centre of the resolution cell, not to the slant range resolution [39]. The causes are uncertainties in system clock timing, jitters in the data sampling clock or uncertain propagation delay of the signal in the atmosphere.

Radar Altitude

The uncertainty σ_{z_1} refers to the uncertainty in radar platform altitude. For airborne InSAR, this uncertainty is of the same order as the baseline uncertainties.

Baseline

The uncertainties σ_b and σ_{θ_b} refer to an uncertainty in the baseline length and orientation [39]. For repeat pass InSAR this is caused by motion compensation errors and uncertainties in the GPS and INS. Flexing of the antenna is not a cause of uncertainty in the SASAR system.

Wavelength

The uncertainty σ_λ refers to the frequency uncertainty of the radar system, due to oscillator instability.

Interferometric Phase

The uncertainty σ_φ refers to an uncertainty in the measurement of the interferometric phase.

4.2.2 Optimum Baseline Elevation Angle

If the baseline elevation angle θ_b is made equal to the antenna look angle at mid swath α_{10} then $\theta_{10} = 0$ and the sensitivity equations show that the height error caused by uncertainty in the baseline length disappears at mid swath [55].

4.2.3 Sensitivity of Parameters

Appendix E contains MathCAD worksheets showing the effect on height error of uncertainties in the SASAR parameters at an operating altitude of 10000 m at mid swath. The effect of pixel posting errors is also shown. The error is proportional to the slope of the terrain [55]. Once the SASAR flightpaths have been set, the height estimation errors due to the various InSAR parameters can be estimated using the MathCAD worksheet.

4.3 Effect of Phase Errors on Baseline Choice

Interferometric phase error is caused by four factors [39]:

- thermal noise
- speckle noise
- pixel misregistration
- geometric or baseline decorrelation

4.3.1 Reducing Phase Errors

The error due to thermal noise can be reduced by increasing the SNR. Speckle noise is reduced by increasing the number of coherent looks of the scene. Pixel misregistration is reduced with more accurate registration algorithms. The effect of pixel misregistration is not discussed further. Geometric or baseline decorrelation between the images depends on the difference in viewing angle on the two flypasts [55].

4.3.2 Geometric Decorrelation and the Critical Baseline

The geometric correlation between two images is given as [39, 55]

$$\alpha = 1 - \frac{Q(\Delta\gamma)(\delta R_s)}{\lambda \tan\gamma_1} = 1 - \frac{Qb(\delta R_s)}{\lambda r_1 \tan(\theta_1 + \theta_b)}$$

This drops as the baseline length increases, thus increasing the phase error.

The overall correlation between the two images is modified by the thermal correlation [55, 67] to become

$$\gamma = \frac{|\alpha|}{1 + \frac{1}{SNR}}$$

The decorrelation is defined as [55]

$$\delta = 1 - \gamma$$

The critical baseline is reached when the geometric correlation is zero. This occurs when

$$b_c = \frac{\lambda r_1 \tan(\theta_1 + \theta_b)}{Q(\delta R_s)}$$

The effect of decorrelation on phase error has been modelled for a varying number of looks using the method outlined by Li and Goldstein [39]. Figure 4.2 shows the phase error against decorrelation for different numbers of looks. As the number of looks increases the phase error approaches the Cramer-Rao bound [55, 39].

4.3.3 Sensitivity of Height Error on Phase Error

The sensitivity of the height error to phase error decreases with increasing baseline, as is shown by the sensitivity equation

$$\sigma_z(\varphi) = \frac{r_1}{b} \frac{\lambda}{Q 2\pi} \frac{\sin(\theta_1 + \theta_b)}{\cos\theta_1} \sigma_\varphi$$

This process opposes the effect of geometric decorrelation with increasing baseline length. An optimum baseline can be found which balances these two terms [39, 55].

4.3.4 Optimum Baseline Length

The two factors can be reconciled optimally by expressing the phase error σ_φ and baseline b written in terms of the correlation. The number of looks chosen is 16 so that the Cramer-Rao bound can be used as an accurate phase error model. The effect of decorrelation on height error is shown in Figure 4.3 for SNR of 10 dB, 20 dB and 80 dB. For the SASAR system the

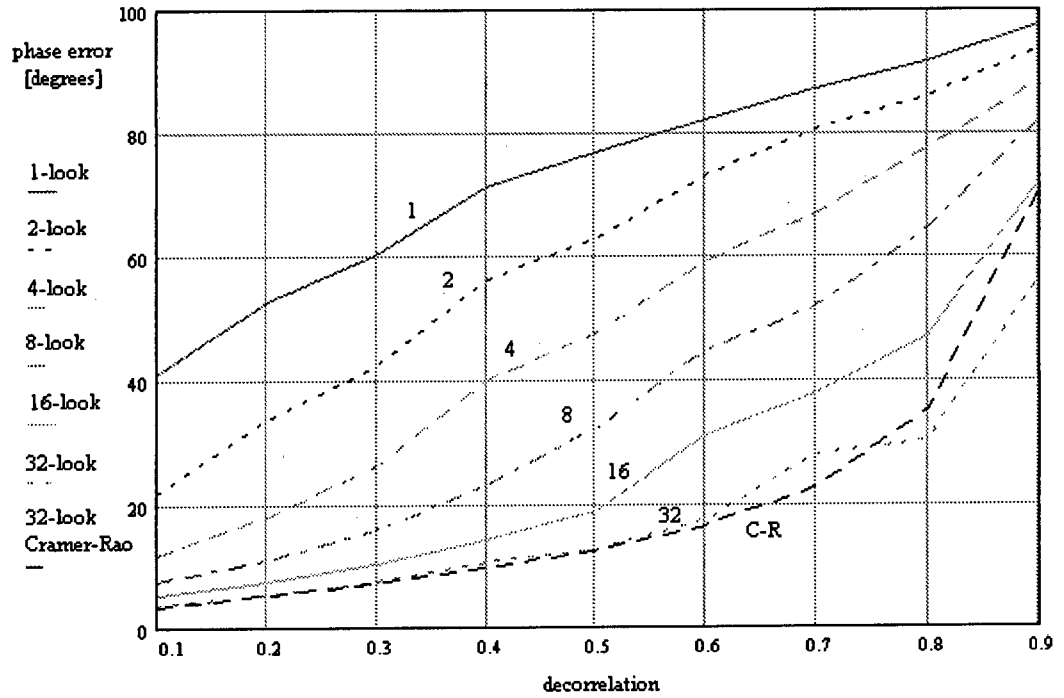


Figure 4.2: Effect of Decorrelation on Phase Error

optimum decorrelation, assuming > 10 dB SNR, is approximately 0.4. However, the shallow gradient of the height error curve means that a decorrelation of between 0.2 and 0.8 can be used. From this decorrelation, the range of acceptable baselines can be calculated.

4.4 SASAR Swath Relationships

Thus section derives equations relating the SASAR altitude and incidence angle.

4.4.1 Incidence Angle Determination

The relationship between incidence angle and radar height for the swath of the SASAR system is given by the following equations. The slant range

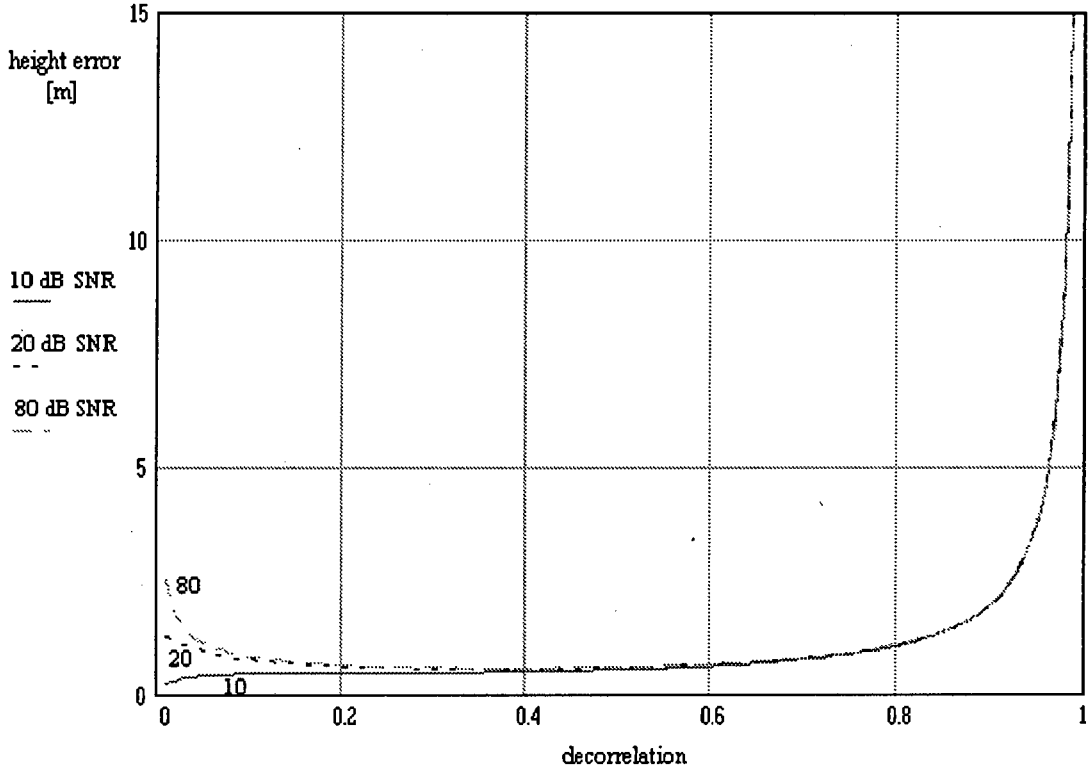


Figure 4.3: Effect of Decorrelation on Height Error

swath width is

$$S_s = 4096 \cdot n_R$$

where n_R is the range sampling interval of 13 m. The near swath look angle is specified as 45°, so for an altitude z_1 the mid swath slant range is

$$r_{10} = \sqrt{2} z_1 + \frac{S_s}{2}$$

or in terms of the incidence angle for flat terrain

$$r_{10} = \frac{S_s}{2} \left(\frac{1}{1 - \sqrt{2} \cos \gamma_f} \right)$$

The local incidence angle is calculated by subtracting the terrain slope θ_t

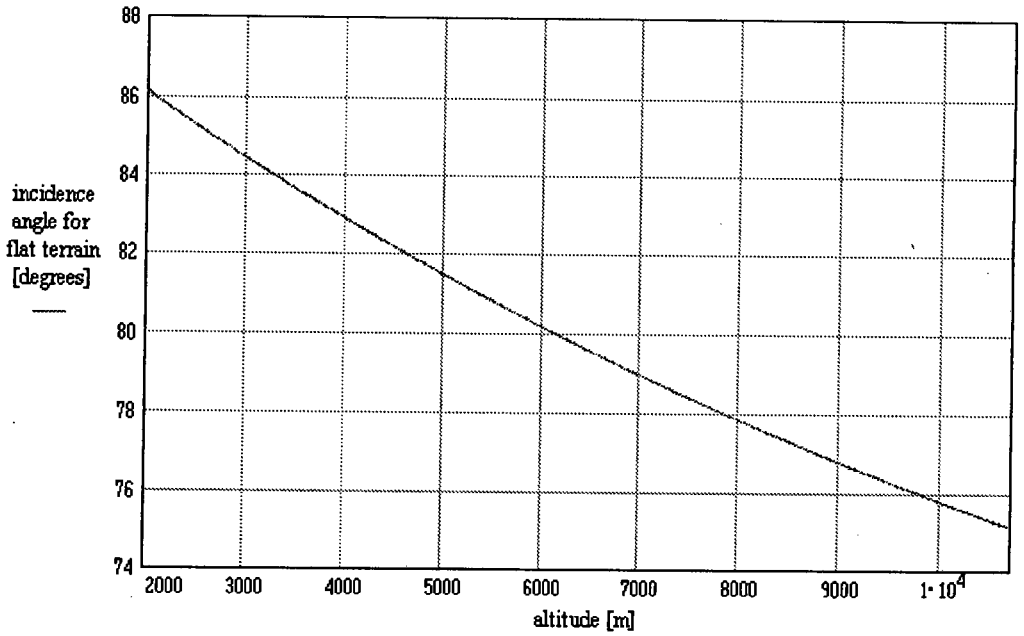


Figure 4.4: Height and Incident Angle Interdependence for SASAR

$$\gamma_l = \gamma_f - \theta_t$$

Figure 4.4 shows the relationship of height and incidence angle for flat terrain for the SASAR system.

4.4.2 Critical Baseline Length Determination

The results of the fringe sampling requirements have to be consistent with the decorrelation baseline length requirements. The cases of a flat surface and a sloped surface are considered. The baseline for a specific decorrelation can be restated in terms of γ_f as

$$b_c = \frac{\lambda r_{10} \tan\gamma_f}{Q(\delta R_s)} = \frac{\lambda \tan\gamma_f}{Q(\delta R_s)} \cdot \frac{S_s}{2} \left(\frac{1}{1 - \sqrt{2}\cos\gamma_f} \right)$$

Figure 4.5 shows the baselines acceptable for decorrelation of 0.2 to 0.8 (see

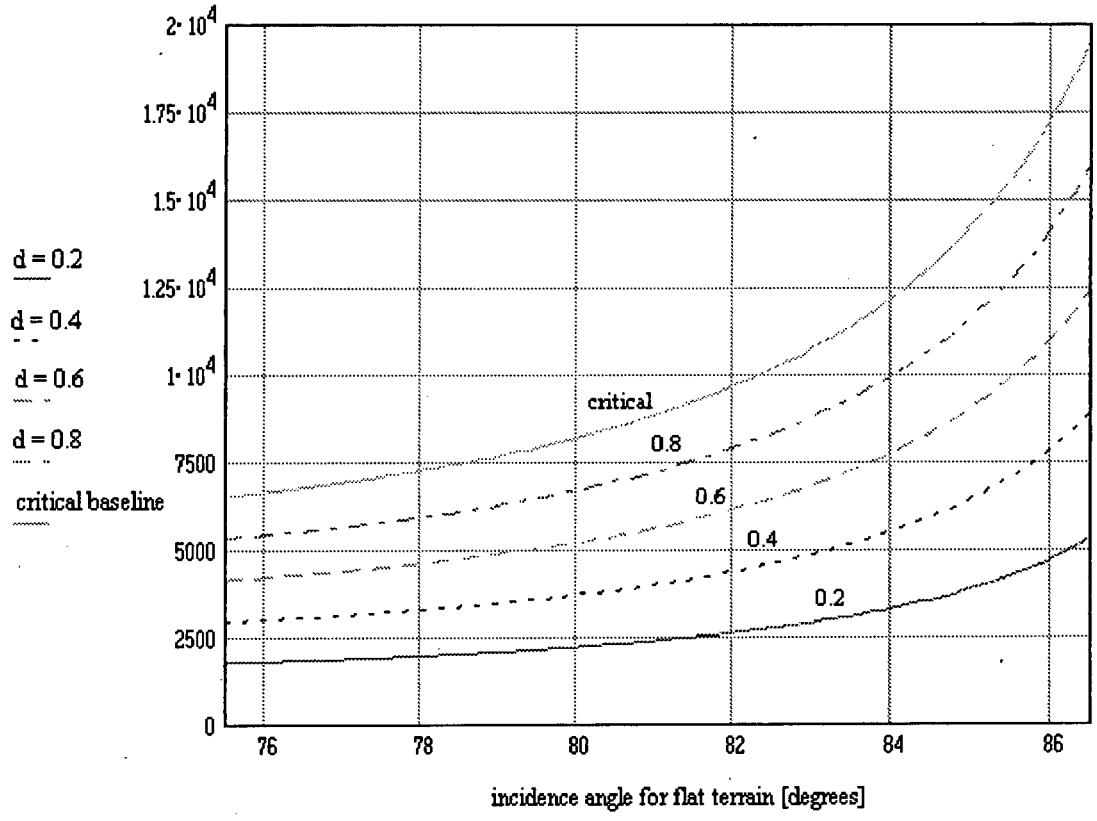


Figure 4.5: The Range of acceptable Baselines for Use with different Decorrelations and Incidence Angles

Figure 4.3) as a function of incidence angle for a 10 dB SNR.

$$b = b_c \frac{1}{1 + \frac{1}{SNR}} \delta$$

4.5 Determination of the Maximum Imagable Terrain Slope

Repeat pass systems can be tailored to maintain a baseline perpendicular to a chosen look direction for a particular type of terrain.

The fringe spacing for the imaged terrain must be such that a fringe in the flat earth subtracted interferometric phase image is sampled at least twice to satisfy the Nyquist criterion. In noisy areas more samples may be required [70, 69].

The phase as discussed in Section 2.3 is given as

$$\varphi = \frac{2\pi Q b_\perp r_1}{\lambda z_1}\left[\frac{1}{\tan\gamma_1} - \frac{1}{\tan\theta_t}\right]$$

The maximum slope θ_t imagable is that at which a 2π interferometric phase difference occurs. The height drop associated with a 2π change in phase is

$$\Delta z = \frac{\lambda z_1 \sin\gamma_f}{Q b_\perp}$$

The terrain slope θ_t is then given by

$$\theta_t = \sin^{-1}\left(\frac{\Delta z}{\Delta r}\right)$$

where Δr is the slant range distance on the image resulting in a height change of Δz. For two samples per fringe,

$$\Delta r = 2n_R$$

Thus, the maximum slope imagable is

$$\theta_{t\max} = \sin^{-1}\left(\frac{\lambda}{2Qn_R}\frac{z_1}{b_\perp}\sin\gamma_f\right)$$

The first term in the bracketed expression is fixed by the physical radar configuration. The second term can be varied, most importantly the ratio $\frac{z_1}{b_\perp}$, which can be varied within two limitations.

Firstly, z_1 is constrained to the operating envelope of the SASAR system, in this case to between 2000 m and 10700 m altitude. Secondly, $b_\perp$ is constrained to between 0.2 and 0.8 of the critical baseline. This in turn varies with incidence angle, or equivalently, altitude, as shown by the derived relationships between incidence angle and altitude for the SASAR system.

Figure 4.6 shows the maximum imagable slopes for the entire altitude range. This requires a mimimum baseline, or $b_\perp = 0.2b_c$. For lesser decorrelations, Figure 4.7 shows the results.

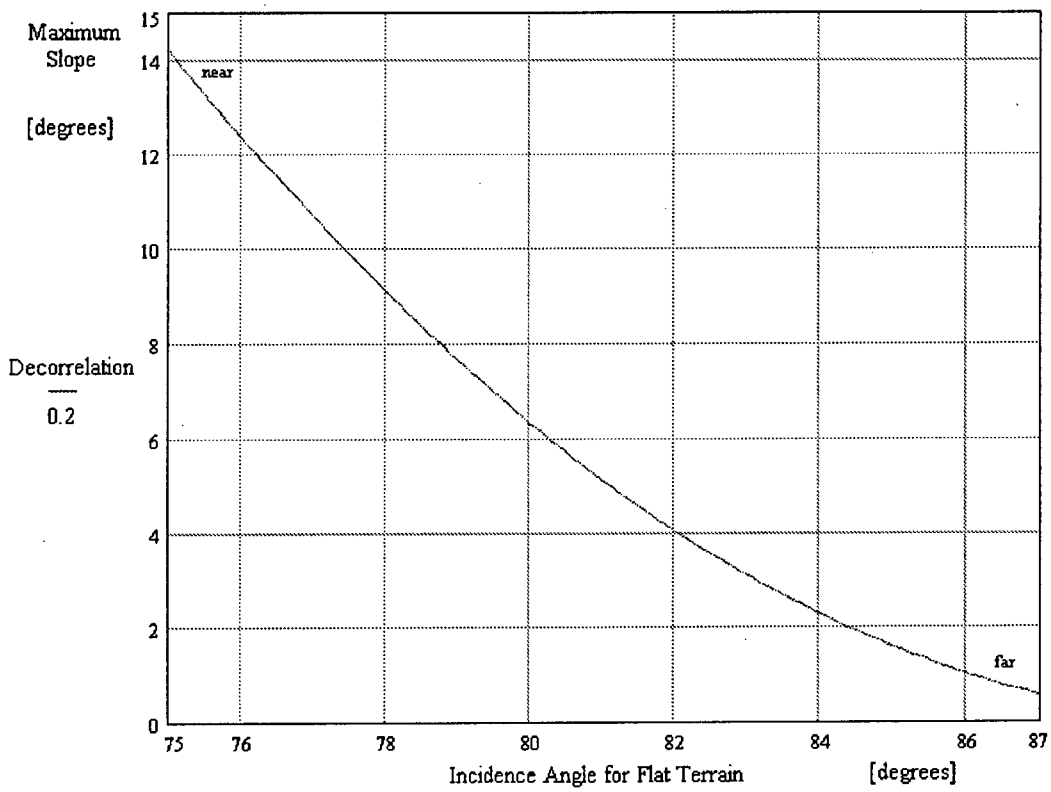


Figure 4.6: Maximum Imagable Slopes versus Incidence Angle for Flat Terrain

4.6 Discussion of Results

The previous discussions are used to evaluate the effectiveness of SASAR-based InSAR for flat surfaces and sloping surfaces.

4.6.1 Baseline Dependency on Incidence Angle

Table 4.2 summarizes the swath constraints for SASAR .The data is obtained from Figures 4.4 and 4.5. For an analysis of maximum imagable slopes, the baseline used is 0.2 of the critical baseline.

The baseline orientation is perpendicular to the look direction, i.e. the baseline is elevated above the horizontal by an angle equal to the incidence

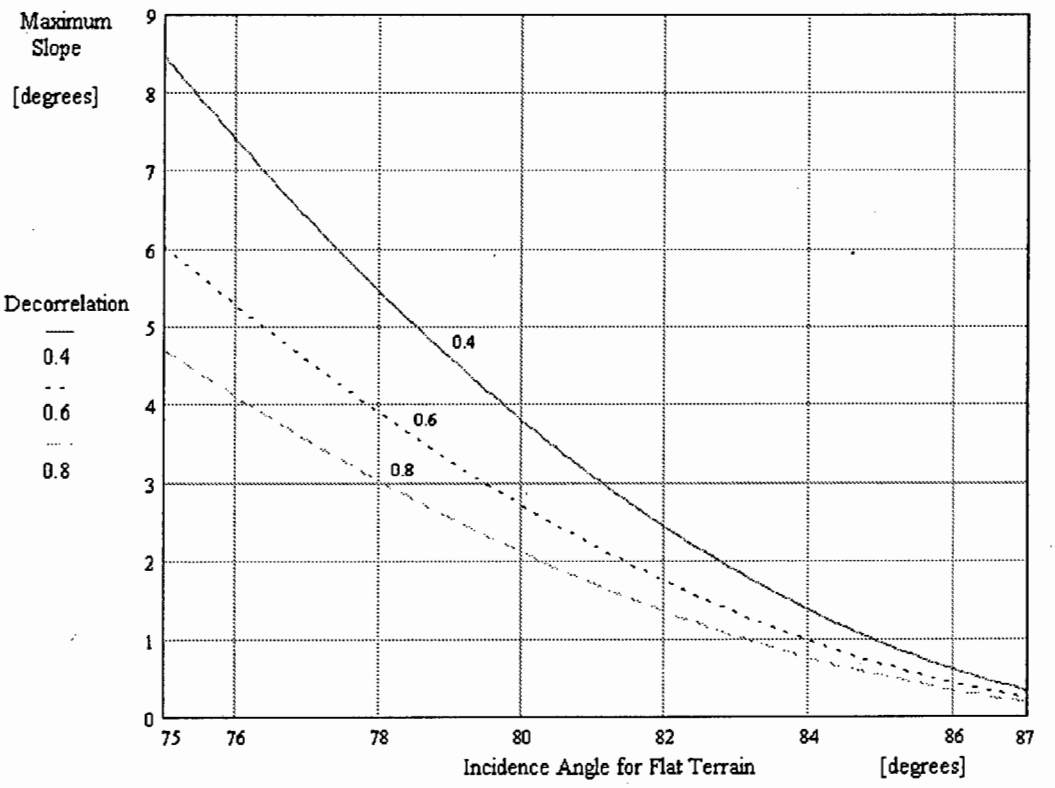


Figure 4.7: Maximum Imagable Slopes versus Incidence Angle for Flat Terrain, for different Decorrelation Values

angle for flat terrain. A pair of horizontal flightpath can be also flown, and the perpendicular baseline calculated from the geometry of the two flight paths.

4.6.2 Results of Analysis

From Figures 4.5, 4.6 and 4.7 the maximum slopes that can be imaged with a given baseline can be found. Table 4.3 lists the slopes that can be mapped with baselines within the decorrelation range of 0.2 to 0.8 at the two extremes of the SASAR operating altitudes. As an be seen, the lower altitudes result in baseline requirements which cannot be met. The range of slopes that can be imaged at mid swath ranges from 0° to 12.3°. The ground swath that is

Table 4.2: SASAR Swath Constraints

Altitude	Incidence Angle	Critical Baseline
10700 m	75.8°	6400 m
6200 m	80°	8300 m
2000 m	86.1°	17600 m

Table 4.3: Maximum Imagable Slopes at a given Incidence Angle

Radar Position	Decorrelation	Baseline	Maximum Slope
10700 m 75.8°	0.2	1900 m	12.3°
	0.4	2950 m	7.7°
	0.8	5500 m	4.2°
2000 m 86.1°	0.2	5400 m	1.0°
	0.4	8750 m	0.6°
	0.8	15800 m	0.3°

useful to InSAR processing from this analysis is from the near swath to mid swath. At further distances, the maximum imagable slope falls to below the maximum at mid swath. However, imaging is still possible if for example a 2° slope is being imaged at further than mid swath by a configuration with a baseline capable of imaging 10° slopes.

4.6.3 Conclusions

The suggested flight paths for the SASAR system to obtain SAR images which can be interferometrically processed are as follows. The first flight path should be made at 10700 m altitude, while the second should be made at an altitude at least 1842 m lower, i.e. at most 8858 m altitude, and at least 466 m further away from the target terrain. This combination of flight paths creates a baseline of at least 1900 m with a normal equal to the swath depression angle at centre swath of 14.2°. The decorrelation for this baseline is 0.2, and is the minimum recommended according to Figure 4.3. Alternatively, a horizontal pair of flightpaths spaced apart by a distance of 7745 m to give a resultant minimum 1900 m perpendicular baseline.

The terrain should be flat, with slopes preferably not exceeding 12° since that is the steepest slope that can be imaged at mid swath. At far swath, only slopes of less steepness than this can be imaged, while at near swath,

slopes of greater steepness can be imaged. Allowance also needs to be made for uncompensated aircraft motion which would drive up the phase error.

Chapter 5

Conclusions and Recommendations

This thesis documents the development and background of InSAR up until May 1994. The theory of both across-track and along-track InSAR is summarized for reference purposes. A variety of derivations of the across-track InSAR radar-target geometry equations is presented. The errors inherent in an InSAR system are discussed qualitatively, and sensitivity equations derived from the geometrical equations are used to quantify the errors arising from uncertainties in InSAR parameter estimates. The concept of optimal baseline elevation angle and optimal baseline length is discussed, and the reasons for phase decorrelation investigated.

A simulation of InSAR for point targets is documented in Appendix A through D. This simulation is carried out to test the understanding of InSAR theory and to experiment with the implementation InSAR processing. The extension of terrain modelling capabilities, both by improving the radar return simulator and by investigating further the factors causing decorrelation in interferometric images, especially topography-induced decorrelation, is important for an InSAR simulation environment. I recommend that a dual approach be taken in studying InSAR: firstly, now that the InSAR theory is accessible through the references in this thesis, an effort must be made to keep abreast of theoretical developments in InSAR processing algorithms, and to continue building up a simulation environment in South Africa; secondly, radar data which can be interferometrically processed must be acquired from local or foreign sources to give InSAR researchers the opportunity to acquaint themselves with the difficulties and issues around InSAR processing.

The SASAR system was analysed in terms of its ability to carry out InSAR missions. The analysis was carried out by specifying the swath dimensions and using the InSAR geometry sensitivity equations to determine optimum system parameters. The swath specifications are used to relate the SASAR geometry to the general InSAR geometrical equations and reduce the degrees of freedom in the analysis. If the swath specifications are changed, the simulation will require changes to the equations relating incidence angle to radar platform altitude, although the methodology of the analysis would remain the same.

The results of the analysis show that the current SASAR configuration is able to map terrain slopes of up to 12° at high level at mid swath. The calculations in the analysis have been done for mid swath, but the near swath is capable of imaging any slopes which the mid swath is capable of. The far swath is only useful if the baseline has the capability of imaging greater slopes than it is already imaging at mid swath. The Figures which have been created using MathCAD are a useful design tool for InSAR analysis, and can be adapted to changing swath specifications by changing the equations used to generate them. The sensitivity of height error to uncertainties in the InSAR parameter measurement is presented in MathCAD worksheet format so that the estimated height errors in a given InSAR configuration can be calculated. The SASAR system should first map flat terrain to create interferograms which can be unwrapped easily and give InSAR researchers a starting point for InSAR processing development. I note that the SASAR specifications were not fixed at the time of writing this thesis, and that if they change before the first flight, it will be necessary to repeat the SASAR analysis after modifying the MathCAD worksheets in Appendix E.

Appendix A

Simulation of across track InSAR for Point Targets

This appendix documents the simulation of across track InSAR for point targets. The purpose of the simulation is to verify the understanding of InSAR theory. The simulation is carried out using MathCAD worksheets and a locally developed radar return signal simulator. Two offset radar flypasts are simulated. The raw data is SAR processed, after which an interferogram is formed by complex multiplying the two focussed SAR images. The interferogram is processed by a height extraction algorithm to yield height information for each ground pixel. The simulation results are contained in Appendix C.

A.1 Point Target Simulation Geometry

A single pass system with a 1900 m baseline was simulated to fly past one point target at a height of 10000 m and closest ground range of 20000 m as shown in Figure A.1.

A.2 Simulation Implementation

The processes carried out in the simulation are shown in Figure A.2. The radar simulation processs are implemented by a set of computer programs. The program flowchart is shown in Figure A.3. Command files are used for

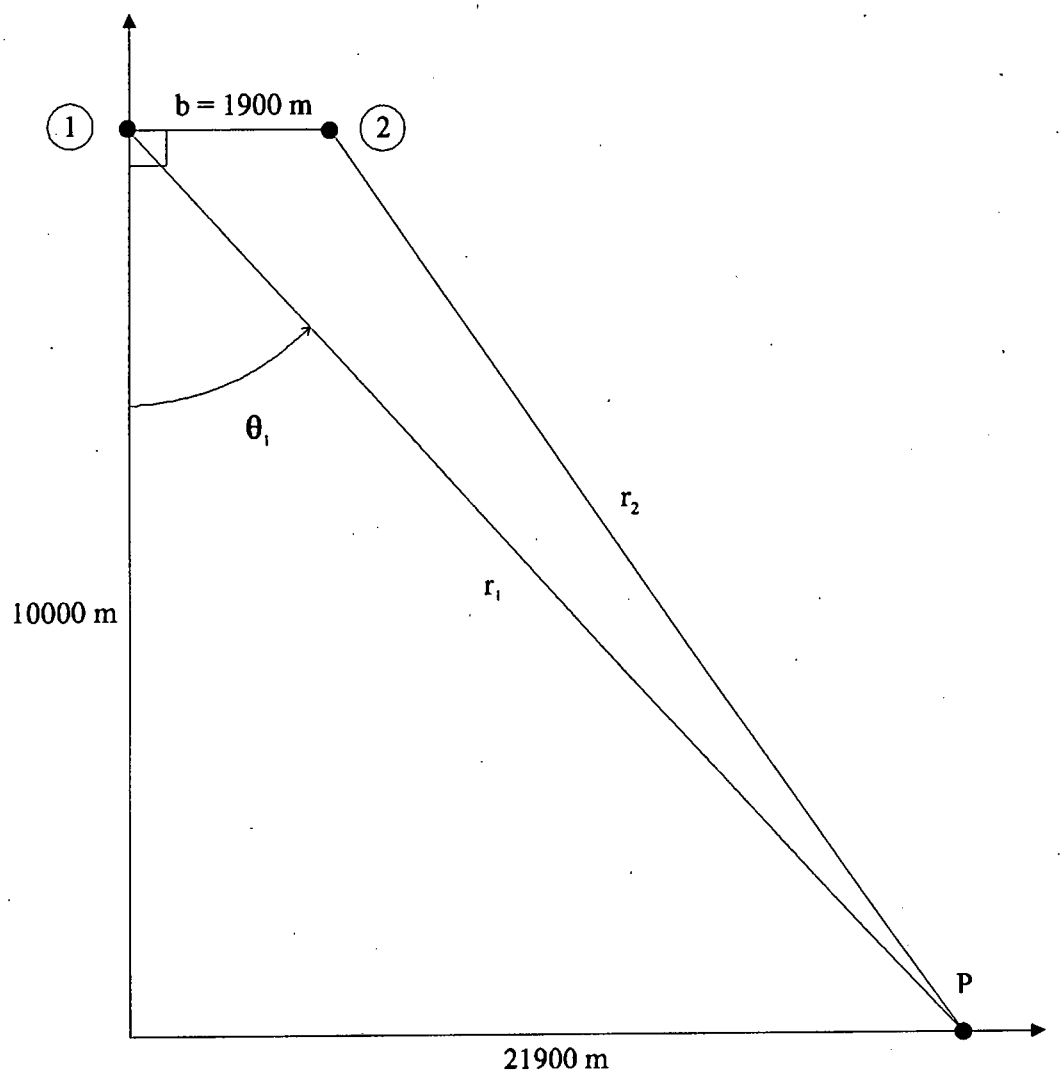


Figure A.1: InSAR Simulation Geometry

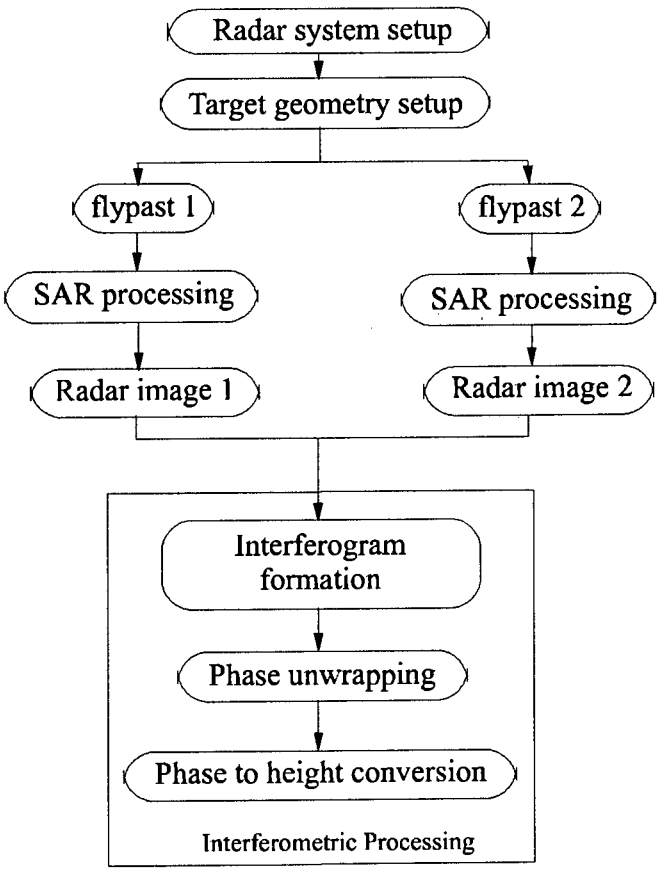


Figure A.2: Simulation Processing Steps

each program to store the necessary parameters, which are precalculated in a MathCAD worksheet.

A.2.1 Simulation Setup

A radar system with specific characteristics and parameters is set up using a MathCAD worksheet. The radar and radar platform specifications, swath and resolution requirements are set and the signal processing requirements necessary to achieve these are determined. The worksheet is contained in Appendix B.

The simulation programs are written in the C programming language, and compiled under DOS using the GNU C compiler to circumvent program memory allocation problems. The program command files are contained in Appendix B while the program listings are contained in Appendix D.

A.2.2 Data Acquisition

RADSIM.EXE

This program is part of an MSc thesis being worked on by Mr. Peter Golda of the RRSG. The program uses information regarding the radar-target geometry, platform flight path, radar pulse and antenna parameters, and range compression parameters to simulate radar pulses and their range compressed returns along the specified flight path [24].

The radar return simulator is run twice using two command files to simulate repeat pass InSAR over the target terrain. The two command files differ in the specification of the radar platform flightpaths.

Program Input:

- sim1.CMD/sim2.CMD

Program Output:

- d1rad.BIN/d2rad.BIN

The images are displayed in Appendix C.

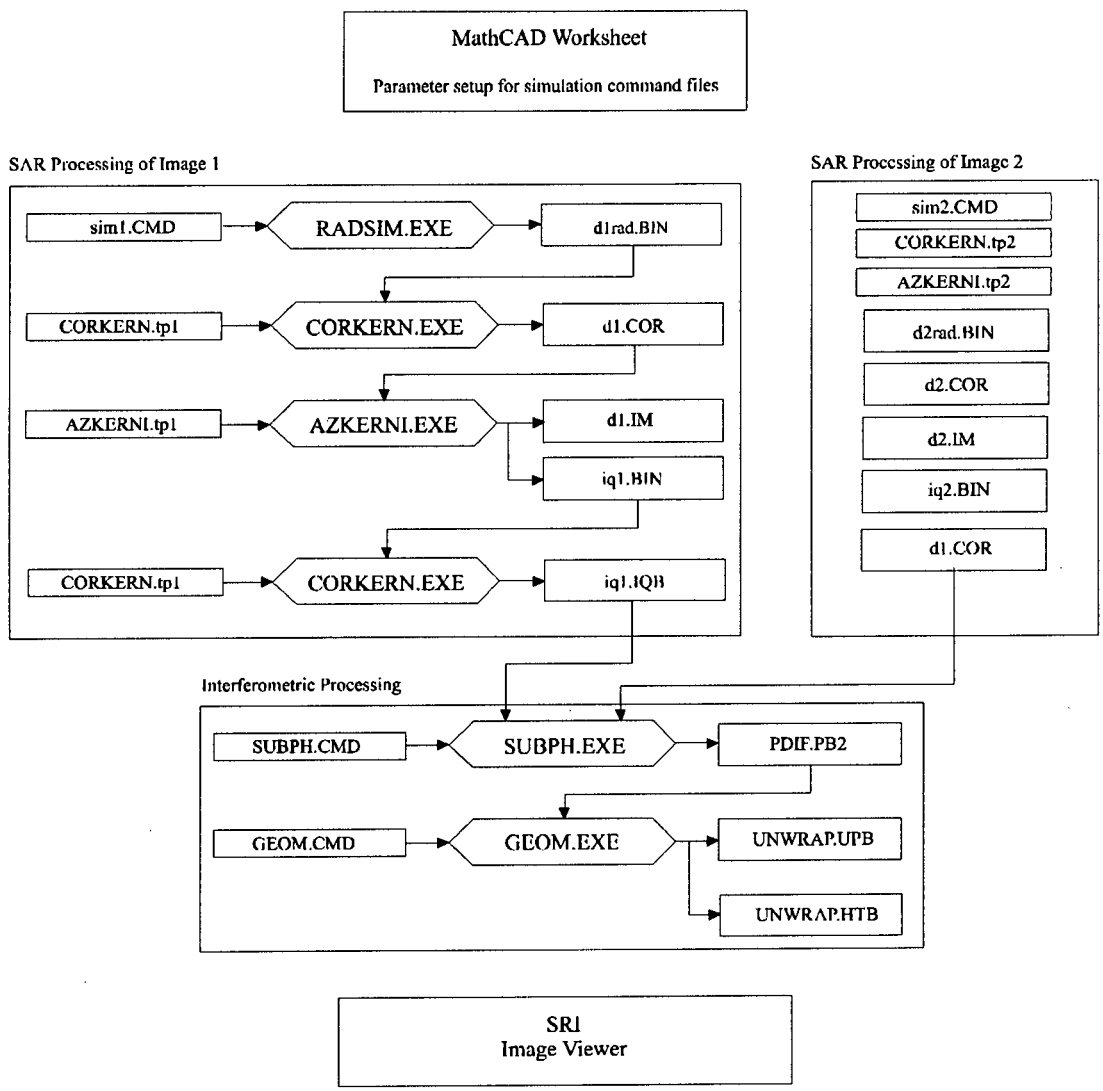


Figure A.3: InSAR Simulation Program Flowchart

A.2.3 SAR Processing

The UCT SAR Processing Kernel was used to focus the SAR images. The kernel programs were developed over a period of two years by members of the RRSG [31, 30].

CORKERN.EXE

This program, which transposes an array, was written by Mr. Jasper Horrell of the RRSG as part of the UCT SAR Processing Kernel. The range focussed data array from the radar simulator is transposed to enable azimuth compression to be done easily by reading sequential memory locations.

Program Input:

- d1rad.BIN/d2rad.BIN
- CORKERN.tp1/CORKERN.tp2

Program Output:

- d1.COR/d2.COR

AZKERNI.EXE

This program is a phase preserving adaptation of an azimuth compression program written by Mr. Jasper Horrell of the RRSG as part of the UCT SAR Processing Kernel. The radar data is focussed in azimuth. the .IMG output file is a magnitude only file, which can be viewed to see the focussed SAR image of a point target. The .BIN file contains the I and Q values of the focussed SAR image, ready to be used in the interferometric processing.

Program Input:

- AZKERNI.tp1/AZKERNI.tp2
- d1.COR/d2.COR

Program Output:

- d1.IM/d2.IM
- iq1.BIN/iq2.BIN

The .BIN images are corner turned again to allow easy array operations on range lines. The data files after corner turning are

- iq1.IQB/iq2.IQB.

These corner turned focussed images are displayed in Appendix C.

IQ2MAG.EXE

This program converts data files consisting of I and Q into magnitude only files. It was written by Mr. Jasper Horrell of the RRSG as part of the UCT SAR Processing Kernel.

A.2.4 Interferometric Processing

Two images of the point targets are acquired using the above documented procedure. The flight paths of the two simulations differ to create a baseline for interferometry. The focussed I and Q values are processed for interferometric processing. No image registration or motion compensation is necessary since the flight paths are known exactly. The programs described below were written specifically for this simulation and are not a part of the UCT SAR Processing Kernal.

SUBPH.EXE

This program complex multiplies the two focussed SAR images to obtain an interferogram. Since the subtraction takes place between corresponding pixels, these must first be located, i.e. the files must be registered. The phase subtraction command file specifies which file to subtract from the other, range and azimuth offsets of one relative to the other in terms of bins, and the bins actually required for processing. This allows small relevant sections of large data files to be extracted for quick processing. The offsets are worked out using the ground range bin size calculated in the setup MathCAD worksheet. It is possible to confirm this result by observing the position of the targets in the two .IMG files when viewing them with the SRI viewing program.

Program Input:

- iq1.IQB

- iq2.IQB
- SUBPH.CMD

Program Output:

- PDIF.PB2

The phase difference image is displayed in Appendix C.

GEOM.EXE

This program converts the differential or wrapped interferometric phase to unwrapped phase and then to height estimates of the terrain. The interferometric phase is unwrapped along each range line, after which, from a knowledge of the ground range to the first pixel and the slant range distance to each bin, the height of each pixel is calculated. The .UPB file contains the unwrapped absolute phase values, while the .HTB file contains the height estimates for each pixel.

Program Input:

- GEOM.CMD
- PDIF.PB2

Program Output:

- UNWRAP.UPA
- UNWRAP.UPB
- UNWRAP.HTA
- UNWRAP.HTB

The unwrapped phase is shown for a few sets of range lines in Appendix C. The height results are also discussed in Appendix C.

A.2.5 Viewing Program

The Shuttle Radar Images (SRI) package is used to view the images. The SRI programs were written in 1993 by Mr. Robert Crida [9, 10].

SRIBLDT.BAT

This batch file calls the program SRIBUILD.EXE with parameters contained in the SRIBUILD.* file to convert input data into a format that can be viewed by the SRI viewing program.

Program Input:

- SRIBUILD.*
- Data file with values in range 0 to 255

Program Output:

- SRI-compatible file

SRI.EXE

This is the shuttle radar image viewer. Any of the radar data files obtained during the simulation can be converted to magnitude images by means of the IQ2MAG.EXE program, further converted to an SRI-compatible format with the aid of the SRIBLDT.BAT file and suitable SRIBUILD.* command files, and viewed to confirm expected results.

A.3 Conclusions

The InSAR simulation for point targets carried out each step from the acquisition of raw data to the extraction of height information. The extracted heights were in agreement with the specified geometry. The interferometric phase could be calculated at the point target location, and no data errors were introduced by the SAR processing.

Appendix B

MathCAD Worksheets and Command Files for the Simulation of InSAR for Point Targets

1. The MathCAD worksheet calculates the parameters for the radar simulator program written by Mr. Peter Golda.

2. The Command files pass parameters on to the executable simulation program files, whose llistings can be found in Appendix D. The use of the command files is described in Appendix A, while the simulation results are contained in Appendix C.

Simulation Parameter Setup

Written by: **Gernot Hassenpflug**
Date: **25 September 1995**

Parameter Listing:

Radar Platform Parameters:

$h := 10000 \cdot m$ altitude of platform

$v := 200 \cdot \frac{m}{sec}$ platform speed

$PRI := 1024$ Number of Pulse Repetition Intervals, i.e. range lines

Antenna Parameters:

$\theta := 50 \cdot deg$ Look angle of antenna to target, from vertical

$\theta_{dep} := \left(-\frac{\pi}{2} + \theta\right)$ depression angle for radsim.exe $\theta_{dep} = -40 \cdot deg$

$f_c := 141 \cdot MHz$ Radar Centre Frequency

$\lambda := \frac{C}{f_c}$ radar wavelength $\lambda = 2.126 \cdot m$

$\alpha_{az} := 8 \cdot deg$ Azimuth beamwidth of real antenna

$\alpha_{el} := 60 \cdot deg$ Elevation beamwidth of real antenna

$l := 1.2 \cdot m$ Length of real antenna

Radar Processing:

Doppler Requirements:

Lower bound on PRF: no Doppler aliasing.
This means that the lower bound is
set by azimuth BW and radar speed

$$BW_{doppler} := \frac{4 \cdot v \cdot \sin\left(\frac{\alpha_{az}}{2}\right)}{\lambda}$$

$BW_{doppler} = 26.247 \cdot Hz$

Reqd: PRF > BWdoppler

for an unlimited BW, azimuth BW = (PRI x az.bin)/Ro

$bins := 512$ Number of range bins per range line

$PRF := 200 \cdot Hz$ Pulse Repetition Frequency

$f_{ad} := 0.01875 \cdot GHz$ A/D Sampling Frequency, equals Simulation Frequency

Pulse Characteristics: $\tau := 170 \cdot nsec$ $B := 18.75 \cdot MHz$ CHIRP

Range Bin size:

$r_{binslant} := \frac{C}{(2 \cdot f_{ad})}$ binsize in slant range $r_{binslant} = 7.994 \cdot m$

$r_{bingnd} := \frac{r_{binslant}}{\sin(\theta)}$ binsize on ground at given look angle $r_{bingnd} = 10.436 \cdot m$

Range Resolution

$\delta_r := \frac{C}{2 \cdot B}$ range resolution $\delta_r = 7.994 \cdot m$

- This should be made to equal range bin size

Azimuth Bin size:

$az_{bin} := \frac{v}{PRF}$ azimuth binsize

Azimuth Resolution

Target:

$RCS := 0.5 \cdot m^2$ radar cross section of target

Higher bound on PRF: no range ambiguity.
This means that elevation BW must be set so that furthest point in swath is < unambiguous range

$r_{un} := \frac{C}{(2 \cdot PRF)}$ $r_{un} = 749.481 \cdot km$

Near and far angles of swath:

$\alpha_n := \theta - \frac{\alpha_{el}}{2}$ $\alpha_n = 20 \cdot deg$

$\alpha_f := \theta + \frac{\alpha_{el}}{2}$ $\alpha_f = 80 \cdot deg$

Slant Range calculations:

According to Antenna specs:

Far: $d_f := \frac{h}{\cos(\alpha_f)}$ $d_f = 57.588 \cdot km$

Near: $d_n := \frac{h}{\cos(\alpha_n)}$ $d_n = 10.642 \cdot km$

Centre: $d_\theta := \frac{h}{\cos(\theta)}$ $d_\theta = 15.557 \cdot km$

has to be < unambiguous range

According to processing requirements:

$R_x := 20000 \cdot m$ closest ground range of target

$\sqrt{d_\theta^2 - h^2} = 1.192 \cdot 10^4 \cdot m$

$R_0 := \sqrt{\left(R_x^2 + h^2\right)}$ closest slant range to target $R_0 = 2.236 \cdot 10^4 \cdot m$

Nearest and Furthest Ground Ranges to process:

$R_f := R_x + \frac{bins}{2} \cdot r_{bingnd}$ $R_n := R_x - \frac{bins}{2} \cdot r_{bingnd}$ $R_\theta := R_x$

$R_f = 2.267 \cdot 10^4 \cdot m$ $R_n = 1.733 \cdot 10^4 \cdot m$ $R_\theta = 2 \cdot 10^4 \cdot m$

has to be smaller than: **has to be greater than:**

$h \cdot \tan(\alpha_f) = 5.671 \cdot 10^4 \cdot m$ $h \cdot \tan(\alpha_n) = 3.64 \cdot 10^3 \cdot m$

Time to and from nearest and furthest processing points:

$t_n := \frac{2 \cdot \sqrt{R_n^2 + h^2}}{C}$ $\sqrt{R_n^2 + h^2} = 2.001 \cdot 10^4 \cdot m$ closest point in swath

$t_n = 1.335 \cdot 10^{-4} \cdot sec$

$t_f := \frac{2 \cdot \sqrt{R_f^2 + h^2}}{C}$ $\sqrt{R_f^2 + h^2} = 2.478 \cdot 10^4 \cdot m$ furthest point in swath

$t_f = 1.653 \cdot 10^{-4} \cdot sec$

Specs needed for RADSIM.EXE program:

Number of Pulse Repetition intervals:	$PRI = 1.024 \cdot 10^3$
Pulse Repetition Frequency:	$PRF = 200 \cdot Hz$
Number of Range bins:	$bins = 512$
Centre frequency:	$f_c = 1.41 \cdot 10^8 \cdot Hz$
Target X-direction off-set:	Choose as 0m
Ground Range at closest approach:	$R_x = 2 \cdot 10^4 \cdot m$
Target height above ground level:	Choose 0 m
Radar cross section of target:	Choose 0.5 m squared
Motion data:	NULL
Aircraft altitude:	$h = 1 \cdot 10^4 \cdot m$
Aircraft speed:	$v = 200 \cdot \frac{m}{sec}$

Rise time: $\tau_r := 1 \cdot nsec$

Pulse Width: $\tau = 1.7 \cdot 10^{-7} \cdot sec$

Fall Time: $\tau_f := 1 \cdot nsec$

Peak Amplitude: Choose 10V

Chirp or Mono: CHIRP

Range Compression can be done with programs externally, or internally with RADSIM.EXE

Radar transmitted power: 1 kW

Centre Frequency: $f_c = 141 \cdot MHz$

Antenna gains: 7 dB

Antenna losses: 0 dB

Radar RCS: 1 m squared

Range offset: Choose so target in centre

Chirp or Mono: CHIRP

Chirp Bandwidth: $B = 18.75 \cdot MHz$

Sine or Null antenna type: Sine

Elevation Beamwidth: $\alpha_{el} = 60 \cdot deg$

Azimuth beamwidth: $\alpha_{az} = 8 \cdot deg$

Depression angle: $\theta_{dep} = -40 \cdot deg$

Fixed squint angle: choose 0 deg

Further factors:

A2D frequency: $f_{ad} = 18.75 \cdot MHz$

UNIT DEFINITIONS:

Base units:

$m \equiv 1L$ $sec \equiv 1T$ $rad \equiv 1$ $deg \equiv \frac{\pi}{180} \cdot rad$

Derived units:

$km \equiv 1000 \cdot m$ $Hz \equiv \frac{1}{sec}$ $kHz \equiv 10^3 \cdot Hz$ $MHz \equiv 10^6 \cdot Hz$ $GHz \equiv 10^9 \cdot Hz$

$msec \equiv 10^{-3} \cdot sec$ $\mu sec \equiv 10^{-6} \cdot sec$ $nsec \equiv 10^{-9} \cdot sec$

$C \equiv 2.997925 \cdot 10^8 \cdot \frac{m}{sec}$

```
! Command File 1 for single target simulation
! Simulation begins

$GENERAL  A
 1024           ! [ ]   Number of Pulse Repetition Intervals to Simulate
  200.00        ! [Hz]  Pulse Repetition Frequency
  512           ! [ ]   Number of range bins to simulate
    0.01875     ! [GHz] Simulation frequency

$TARGET   A
       0.00     ! [m]   Target x direction offset from mid path
   20000.00     ! [m]   Ground range at closest approach
       0.00     ! [m]   Target height above ground level
       0.50     ! [m^2] Target radar cross section

$GEOMETRY G1
          A
       NULL     ! Use which motion data ?
  10000.00      ! [m]   Aircraft altitude
    200.00      ! [m/s] Aircraft velocity

!$WRITE G1, ASC, OVR, K:\GERNOT\FINAL\G1.ASC

$PULSEGEN T1 A1 ! Generate pulse waveform data and put into array A1 and T1
         A      ! Use which GENERAL data ?
       1.0      ! [ns]  Rise time
       1.0      ! [ns]  Fall time
     170.0      ! [ns]  Pulse width
      10.0      ! [V]   Peak amplitude
      CHIRP     ! CHIRP or MONO
       0.01875 ! [GHz] Centre Frequency

!$WRITE T1, ASC, OVR, K:\GERNOT\FINAL\T1.ASC
!$WRITE A1, ASC, OVR, K:\GERNOT\FINAL\A1.ASC

$FFT F1 T1

!$WRITE F1, ASC, OVR, K:\GERNOT\FINAL\F1.ASC

$MATCHFILT F2  ! Output
       F1      ! Input
       1.0     ! Windowing Factor

!$WRITE F2, ASC, OVR, K:\GERNOT\FINAL\F2.ASC

$RETURNL T2 G1 A1
    1.0         ! [kW]  Radar Transmitted Power
    0.141       ! [GHz] Radar centre frequency
    7.0         ! [dB]  Antenna gains
    0.0         ! [dB]  Losses ( TxLoss + RxLoss + PropLoss )
    1.0         ! [m^2] Radar cross sectional area
22304.00        ! [m]   Range offset for geometry setup
 CHIRP          ! Chirp or Mono
    0.01875     ! [GHz] Centre Frequency
  SIN           ! SIN or NULL antenna type
   60.0         ! [deg] Elevation beamwidth
    8.0         ! [deg] Azimuth beamwidth
  -40.0         ! [deg] Depression angle  (down is negative)
    0.0         ! [deg] Fixed squint angle  (forward is positive)

!$WRITE T2, ASC, APP, K:\GERNOT\FINAL\T2.ASC

$AMPLIFY T2 T2 10.0

!$WRITE T2, ASC, APP, K:\GERNOT\FINAL\T2AMP.ASC

$FFT F3 T2

$MULTIPLY F4 F3 F2

$FFT T3 F4

!$WRITE T3, ASC, APP, K:\GERNOT\FINAL\T3.ASC

$A2D D1 T3
  0.00001       ! [V] LSB value
  8             ! [bits] number of bits of A2D precision ( 4, 8 or 16 )
  0             ! [bin]  Starting bin ( inclusive )
  0             ! [bins] How many bins to process after starting bin
                !        ( 0 means process until end of range line
  1             ! [bins/sample] take one sample for how many bins ?

!$WRITE D1, ASC, APP, K:\GERNOT\FINAL\D1RAD.ASC
$WRITE D1, BIN, APP, K:\GERNOT\FINAL\D1RAD.BIN

$ENDLOOP
! End of simulation
```

```
! Command File 1 for single target simulation
! Simulation begins

$GENERAL  A
 1024           ! [ ]   Number of Pulse Repetition Intervals to Simulate
  200.00        ! [Hz]  Pulse Repetition Frequency
  512           ! [ ]   Number of range bins to simulate
    0.01875     ! [GHz] Simulation frequency

$TARGET   A
       0.00     ! [m]   Target x direction offset from mid path
   21900.00     ! [m]   Ground range at closest approach
       0.00     ! [m]   Target height above ground level
       0.50     ! [m^2] Target radar cross section

$GEOMETRY G1
          A
      NULL      ! Use which motion data ?
  10000.00      ! [m]   Aircraft altitude
    200.00      ! [m/s] Aircraft velocity

!$WRITE G1, ASC, OVR, K:\GERNOT\FINAL\G1.ASC

$PULSEGEN T1 A1 ! Generate pulse waveform data and put into array A1 and T1
         A      ! Use which GENERAL data ?
       1.0      ! [ns]  Rise time
       1.0      ! [ns]  Fall time
     170.0      ! [ns]  Pulse width
      10.0      ! [V]   Peak amplitude
      CHIRP     ! CHIRP or MONO
       0.01875  ! [GHz] Centre Frequency

!$WRITE T1, ASC, OVR, K:\GERNOT\FINAL\T1.ASC
!$WRITE A1, ASC, OVR, K:\GERNOT\FINAL\A1.ASC

$FFT F1 T1

!$WRITE F1, ASC, OVR, K:\GERNOT\FINAL\F1.ASC

$MATCHFILT F2  ! Output
       F1      ! Input
       1.0     ! Windowing Factor

!$WRITE F2, ASC, OVR, K:\GERNOT\FINAL\F2.ASC

$RETURNL T2 G1 A1
    1.0         ! [kW]  Radar Transmitted Power
    0.141       ! [GHz] Radar centre frequency
    7.0         ! [dB]  Antenna gains
    0.0         ! [dB]  Losses ( TxLoss + RxLoss + PropLoss )
    1.0         ! [m^2] Radar cross sectional area
22304.00        ! [m]   Range offset for geometry setup
 CHIRP          ! Chirp or Mono
    0.01875     ! [GHz] Centre Frequency
  SIN           ! SIN or NULL antenna type
   60.0         ! [deg] Elevation beamwidth
    8.0         ! [deg] Azimuth beamwidth
  -40.0         ! [deg] Depression angle  (down is negative)
    0.0         ! [deg] Fixed squint angle  (forward is positive)

!$WRITE T2, ASC, APP, K:\GERNOT\FINAL\T2.ASC

$AMPLIFY T2 T2 10.0

!$WRITE T2, ASC, APP, K:\GERNOT\FINAL\T2AMP.ASC

$FFT F3 T2

$MULTIPLY F4 F3 F2

$FFT T3 F4

!$WRITE T3, ASC, APP, K:\GERNOT\FINAL\T3.ASC

$A2D D1 T3
  0.00001       ! [V] LSB value
  8             ! [bits] number of bits of A2D precision ( 4, 8 or 16 )
  0             ! [bin]  Starting bin ( inclusive )
  0             ! [bins] How many bins to process after starting bin
                !        ( 0 means process until end of range line
  1             ! [bins/sample] take one sample for how many bins ?

!$WRITE D1, ASC, APP, K:\GERNOT\FINAL\D2RAD.ASC
$WRITE D1, BIN, APP, K:\GERNOT\FINAL\D2RAD.BIN

$ENDLOOP
! End of simulation
```

(Long integers used throughout)

STARTP	[Index of start pulse to be used (long int)]	==> 0
PRESUM	[Corner prog presum ratio (skips)(long int)]	==> 1
PULSES	[No. of pulses to be used (after presumming)]	==> 1024
HEADER	[Bytes taken up by header]	==> 0
SPACER	[Bytes at end of each range line]	==> 0
MBINS	[Max possible no. of bins]	==> 512
STARTB	[Index of start range bin]	==> 0
NBINS	[Number of range bins (per range segment)]	==> 512
INNAME	[Name of input raw data file]	==> d1rad.bin
OUTNAME	[Output file name]	==> d1.cor
APPEND	[Append existing output file (Y/N)?]	==> N
ITERAT	[Number of range segments (start range bin is incremented by NBINS for each range segment)]	==> 1

(Long integers used throughout)

STARTP	[Index of start pulse to be used (long int)]	==> 0
PRESUM	[Corner prog presum ratio (skips)(long int)]	==> 1
PULSES	[No. of pulses to be used (after presumming)]	==> 1024
HEADER	[Bytes taken up by header]	==> 0
SPACER	[Bytes at end of each range line]	==> 0
MBINS	[Max possible no. of bins]	==> 512
STARTB	[Index of start range bin]	==> 0
NBINS	[Number of range bins (per range segment)]	==> 512
INNAME	[Name of input raw data file]	==> d2rad.bin
OUTNAME	[Output file name]	==> d2.cor
APPEND	[Append existing output file (Y/N)?]	==> N
ITERAT	[Number of range segments (start range bin is incremented by NBINS for each range segment)]	==> 1

URATE	[Screen update rate]	==> 1
DELAY0	[Time delay to input data first range bin (secs)]	==> 148.7e-06
FC	[Carrier frequency]	==> 141.0e+06
PRF	[PRF of input data (Hz) (double)]	==> 200.00
GRNDSPD	[Average ground speed (m/s)]	==> 200.0
MAZPTS	[No. of az pts (I&Q pairs) in inp file (long int)	==> 1024
STARTAZ	[Start az pt to process]	==> 0
AZPTS	[No. of az pts to process]	==> 1024
MBINS	[No. of range bins in input file]	==> 512
STARTBIN	[Start rng bin pt to process]	==> 0
NBINS	[No. of rng bins to process]	==> 512
DCOFF	[DC offset (long int) of input data]	==> 127
FFTPTS	[Azimuth FFT size (powerof 2)]	==> 8192
INNAME	[Input file name (before presumming)]	==> d1.cor
OUTNAME	[Output file name (image file)]	==> d1.im
FOCSTEPS	[No. rng focus updates (in proc segment)]	==> 4
REFSIGN	[Phase runout sign for reference function (+-1)	==> -1
A2D	[A/D Sampling frequency (Hz)]	==> 0.01875e+09
AZRES	[Required azimuth resolution (m)]	==> 8.0
WINC	[Window constant - time domain (double)]	==> 0.08
POSTRAT	[Post summing ratio]	==> 1
SCALE	[Scale factor for output magnitude (double)]	==> 60

URATE	[Screen update rate]	==> 1
DELAY0	[Time delay to input data first range bin (secs)]	==> 148.7e-06
FC	[Carrier frequency]	==> 141.0e+06
PRF	[PRF of input data (Hz) (double)]	==> 200.00
GRNDSPD	[Average ground speed (m/s)]	==> 200.0
MAZPTS	[No. of az pts (I&Q pairs) in inp file (long int)	==> 1024
STARTAZ	[Start az pt to process]	==> 0
AZPTS	[No. of az pts to process]	==> 1024
MBINS	[No. of range bins in input file]	==> 512
STARTBIN	[Start rng bin pt to process]	==> 0
NBINS	[No. of rng bins to process]	==> 512
DCOFF	[DC offset (long int) of input data]	==> 127
FFTPTS	[Azimuth FFT size (powerof 2)]	==> 8192
INNAME	[Input file name (before presumming)]	==> d2.cor
OUTNAME	[Output file name (image file)]	==> d2.im
FOCSTEPS	[No. rng focus updates (in proc segment)]	==> 4
REFSIGN	[Phase runout sign for reference function (+-1)	==> -1
A2D	[A/D Sampling frequency (Hz)]	==> 0.01875e+09
AZRES	[Required azimuth resolution (m)]	==> 8.0
WINC	[Window constant - time domain (double)]	==> 0.08
POSTRAT	[Post summing ratio]	==> 1
SCALE	[Scale factor for output magnitude (double)]	==> 40

```
! SUBPH.CMD command file for SUBPH.C program

! Simulation directives

#ASCFILE    Y
#OUTNAME    pdif
#UPRATE     5

! INTINIT Parameters

NPRIS     >   1024        ! Number of PRI intervals
BINS      >   512         ! Number of range bins
R_OFF     >   214         ! Range bin offset of files, file 1 being origin, 214
                          ! R-OFF being the number of bins added to file 2's
                          ! bin numbers to shift target to position in file 1
A_OFF     >   0           ! Azimuth offset
S_R_BIN   >   0           ! First range bin to be processed in file 1
S_A_BIN   >   0           ! First azimuth bin
N_R_BINS  >   16          ! Number of Range bins to process 16
N_A_BINS  >   1024        ! Number of Azimuth bins to process

! 1st File (near path) has bigger numbers hopefully!!!
%
INFILE    >   iq1.iqb

! 2nd File (far file) has smaller values with some luck!!!
%
INFILE    >   iq2.iqb

! File ends
```

```
! GEOM.CMD command file for GEOM.C program

! Simulation directives

#ASCFILE    Y
#OUTNAME    unwrap
#UPRATE     5

! FINDHGT Parameters

NPRIS     >   1024        ! Number of PRI intervals
BINS      >   16          ! Number of range bins
S_R_BIN   >   0           ! First range bin to be processed in file 1
S_A_BIN   >   0           ! First azimuth bin
N_R_BINS  >   16          ! Number of Range bins to process
N_A_BINS  >   1024        ! Number of Azimuth bins to process

PASSES    >   2           ! Number of flypasts: one or two
V         >   200         ! aircraft speed                        [m/s]
HGT       >   10000       ! Altitude of aircraft                  [m]
BL        >   1900        ! Interferometric Baseline length       [m]
ALPHA     >   0           ! Baseline tilt above the horizontal    [degrees]
A2D       >   1.0e+9      ! A2D Frequency                         [Hz]
TD        >   148.7e-6    ! Time delay to nearest range bin       [secs]
LAMBDA    >   2.126       ! Wavelength                            [m]

! Interferometric phase file
%
INFILE    >   pdif.pb2

! File ends
```

Appendix C

Point Target InSAR Simulation Results

This appendix contains the results of the InSAR point target simulation described in Appendix A. The images are:

- range compressed SAR images output from RADSIM.EXE
- azimuth and range focussed SAR images after SAR Kernel processing
- phase difference image
- unwrapped phase for a selection of range lines

1. Figure C.1 represents file 'd1rad.BIN'. The aircraft flies upwards on the left of the image, parallel to the target line.
2. Figure C.2 represents file 'd2rad.BIN'. The aircraft flies upwards on the left of the image, parallel to the target line.
3. Figure C.3 represents file 'd1.IM', the fully focussed SAR image. The aircaft flight path is along the top of the image, towards the right.
4. Figure C.4 represents file 'd2.IM', the fully focussed SAR image. The aircaft flight path is along the top of the image, towards the right.
5. Figure C.5 represents file 'iq1.IQB', the same as 'd1.IM' but derived from I and Q values via IQ2MAG.EXE.

6. Figure C.6 represents file 'iq2.IQB', the same as 'd2.IM' but derived from I and Q values via IQ2MAG.EXE.

7. Figure C.7 represents file 'pdif.PB2'. The aircraft flies up along the left side of the image.

8. Figure C.8 shows the phase unwrapping of the first four range lines. The phase which has been unwrapped is an offset zero and not an interferometric phase.

The height extracted from the simulation was 9638.24 m ±0.03 m over the entire area. These results were achieved not by unwrapping the phase, since this cannot be done when there is no surfacem but by calculation using the distances to each range bin in the case of a flat terrain to obtain the expected interferometric phase. Rounding errors in the C programming language trigonometric function leads to the discrepancy between the experimentally derived value and the actual height of 10000 m.

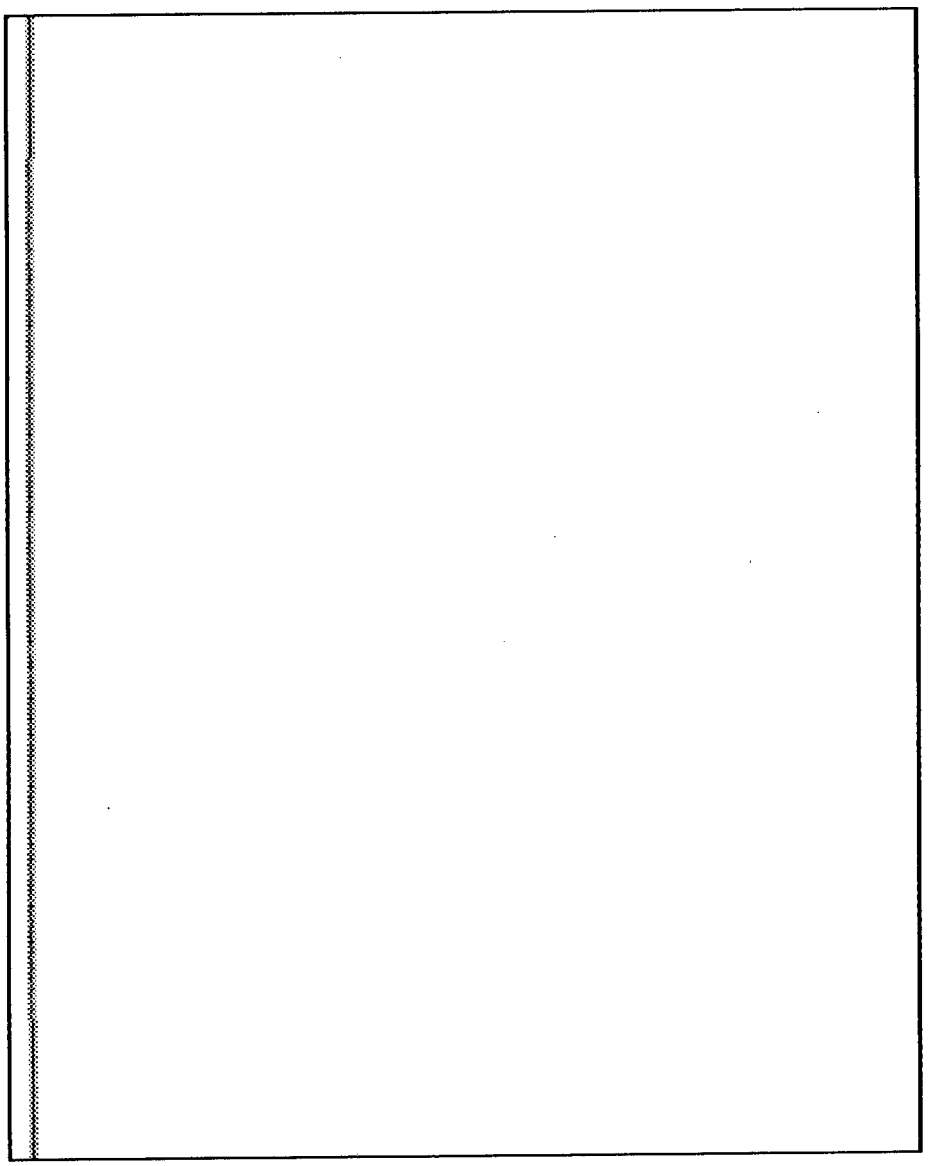


Figure C.1: Target seen on Near Flight Path after Range Compression

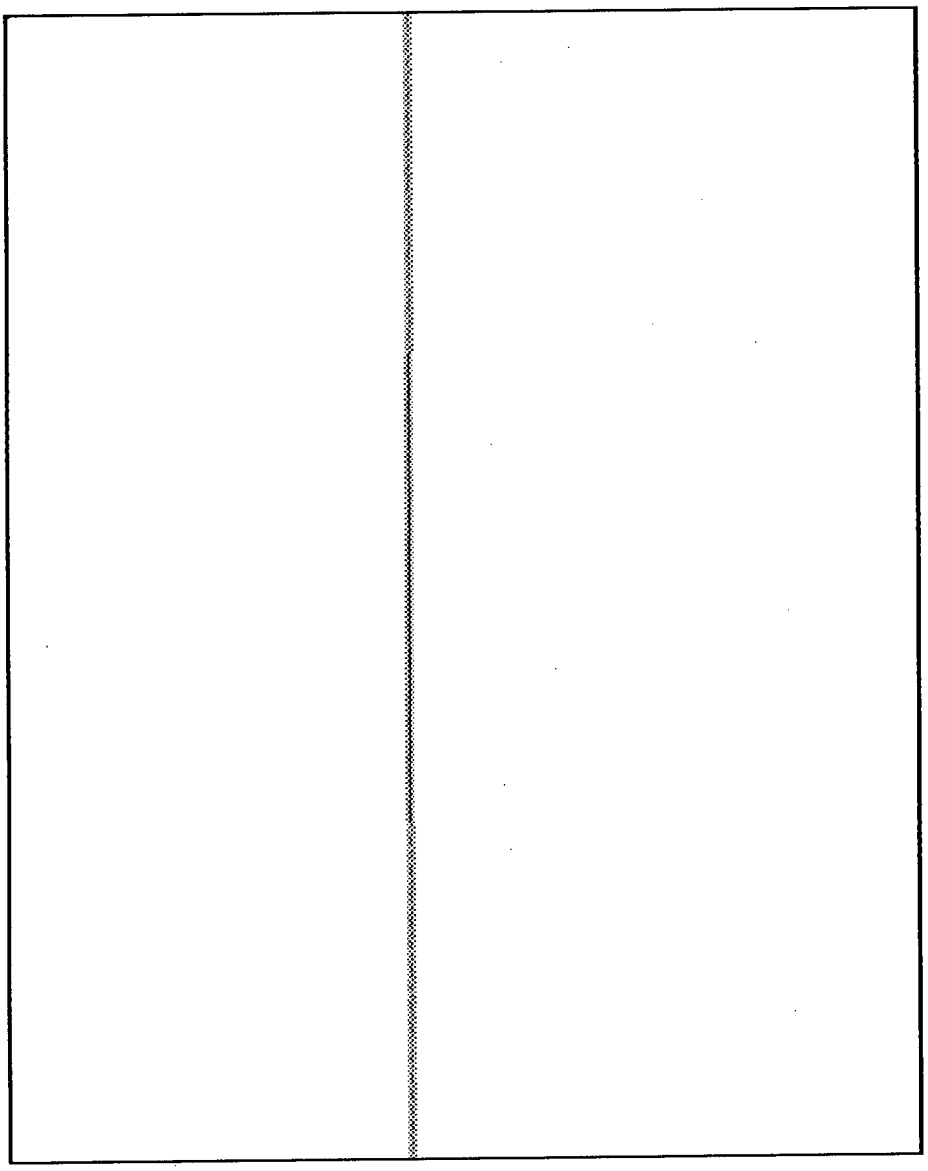


Figure C.2: Target seen on Far Flight Path after Range Compression

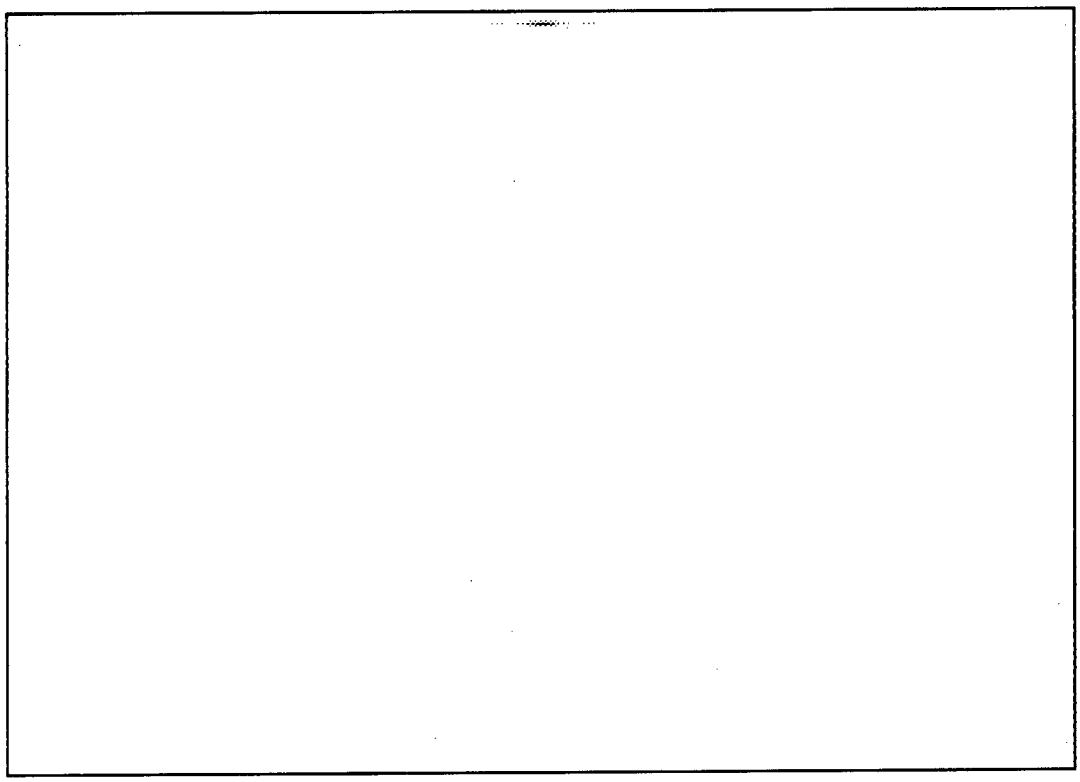


Figure C.3: Corner turned Image showing Target on Near Flight Path after Azimuth Focussing

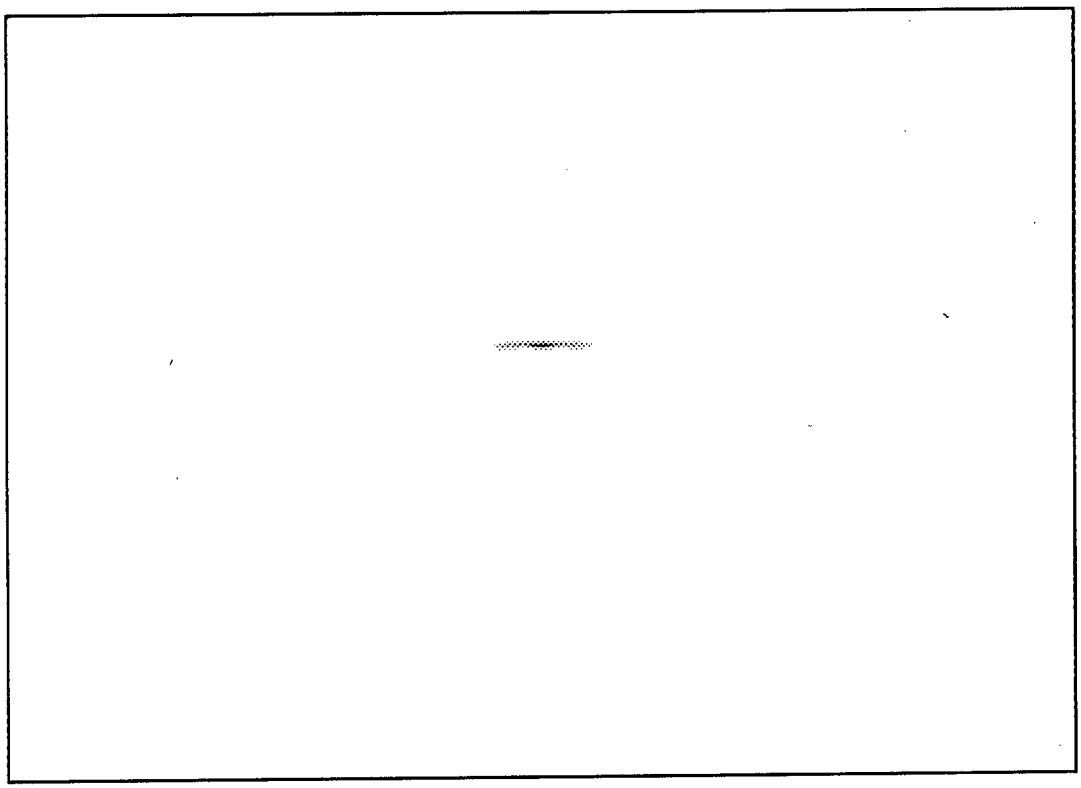


Figure C.4: Corner turned Image showing Target on Far Flight Path after Azimuth Focussing

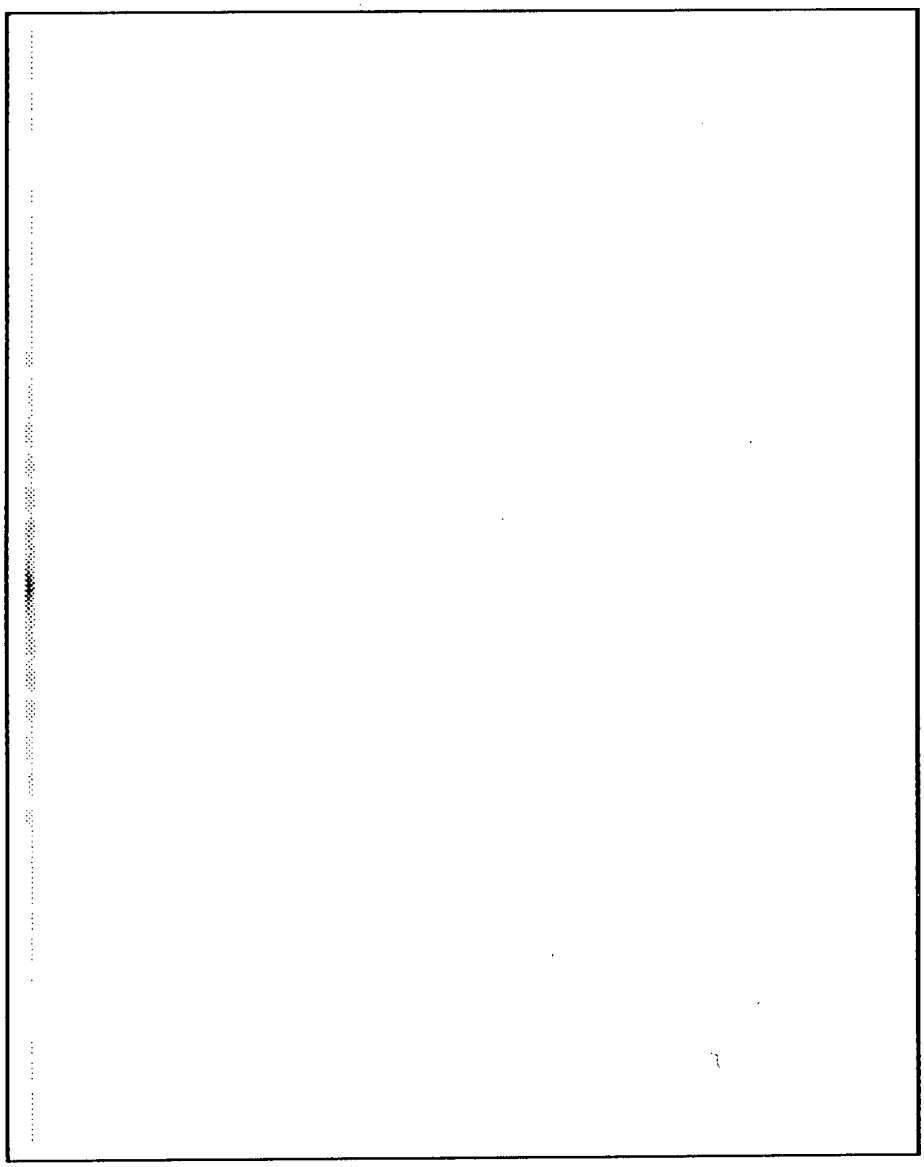


Figure C.5: Magnitude Image constructed from I and Q values from the Output of the Azimuth Compression Program showing the Target on the Near Flight Path

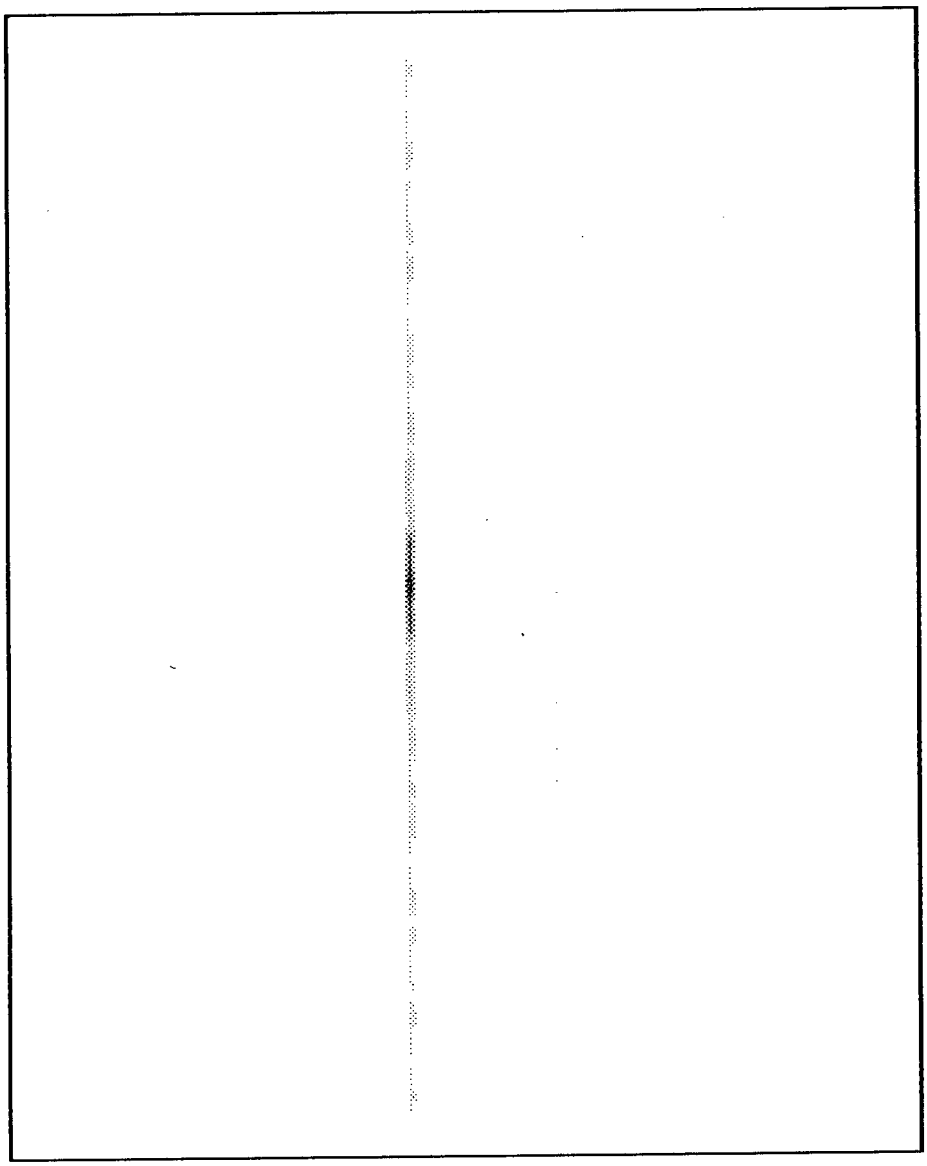


Figure C.6: Magnitude Image constructed from I and Q values from the Output of the Azimuth Compression Program showing the Target on the Far Flight Path

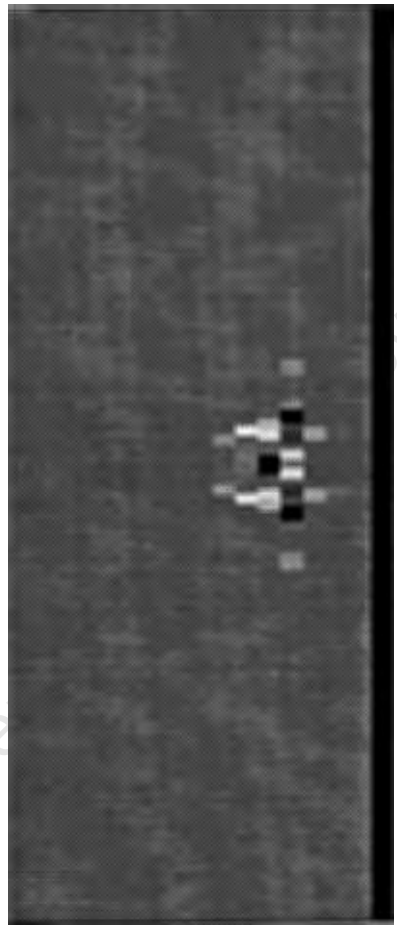


Figure C.7: Phase Difference for a Point Target

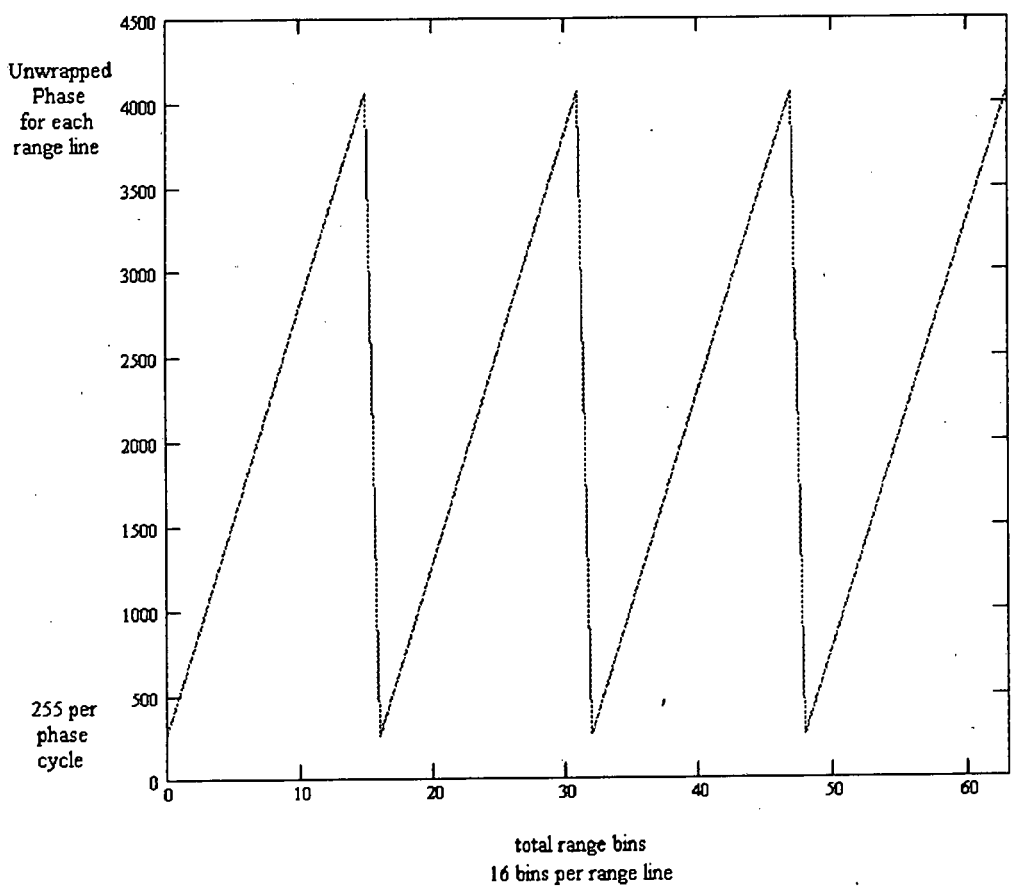


Figure C.8: The Unwrapped Phase over the first four Range Lines

Appendix D

Listings of the Program Code for the InSAR Simulation for Point Targets

The following programs are listed. Their use is described in Appendix A.

1. SUBPH.C
2. GEOM.C

```
/*************************************************************************/
/* Program to generate interferometric phase of 2 files; ie. with Q      */
/* values subtracted to give the phase difference of data points.        */
/* --------------------------------------------------------------------- */
/* Original Author     :  Gernot Hassenpflug                             */
/* Date                :  16 October 1995                                */
/* Modified            :  25 September 1996                              */
/* Program listing     :  SUBPH.C                                        */
/* Set up requirements :  SUBPH.CMD, two I & Q data files                */
/* Output destination  :  <filename>.PHB, .PHA, .PB2                     */
/*************************************************************************/

/* Include library functions */

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

/* Type definitions */

typedef unsigned char string[14]; /* string definition for filenames    */

/* Declare file pointers */

FILE   *cmdfile,          /* The command file "SUBPH.CMD"                */
       *datfile,          /* The data file for I & Q output              */
       *datfile2,         /* Data file for 0-255 output                  */
       *indatfile;        /* The data file for input                     */

/* Variables assigned values from command file  */


int    NPRIS,             /* No. of pulse repetition intervals    [none] */
       BINS,              /* Number of range bins                 [none] */
       R_OFF, A_OFF,      /* Offsets for range and azimuth               */
       S_R_BIN,S_A_BIN,   /* Starting bins in range and azimuth          */
       N_A_BINS,N_R_BINS;/* Number of bins to read                       */

string *FNAME;            /* pointer to array of input filenames         */

/* Variables assigned values within program */

int    test,              /* Store for character read from command file  */
       line_count,        /* Index to line number of command file        */
       fil_index,         /* Index to the file number being processed    */
       update_rate,       /* Progress report every n-th iteration        */
       valid_run,         /* Indicates whether a file name was defined   */
       ascii_file;        /* Indicates whether an ASCII file is required */

int    variable_log[16], /* Flags indicating variables defined          */
       max_val;           /* Maximum I or Q value in digital form        */

unsigned char *phasevals;   /* Array for output phase in 0-255 range     */
unsigned char *I_VALS;      /* Array for input I values                  */
unsigned char *Q_VALS;      /* Array for input Q values                  */
unsigned long int q_index,  /* Index to the Q value being processed      */
                  iq_index; /* I or Q value in input files               */

float *P_VALS;            /* Array for output phase values               */
```

```
/* Function Prototypes - in the order that they appear in the code */

  /* Functions concerning command file reading and error checking. */
  void error_initialize( void );
  void exit_proc( void );
  void parse_directive( void );
  void parse_variable( void );

  /* Functions to do with the actual simulation and the log file */
  void INTINIT( void );
  void store( void );
  void Shift( unsigned long int );
  void Read( int );
  void Readsub( int );
  float Angle( float, float );


/*************************************************************************/
/* Main Program                                                          */
/*************************************************************************/

main()
{
  /* Initialize error messages */
  error_initialize();

  /* Initialize variables and flags */
  line_count = valid_run = ascii_file = 0;
  fil_index = -1; update_rate = 100; max_val = 0;
  for (test=0; test<16; variable_log[test++]=0);

  /* Print headings on the screen */
  /* system("cls");*/
  /* clrscr(); */
  puts("\t THIS PROGRAM FINDS THE INTERFEROMETRIC PHASE OF 2 SPECIFIED\n");
  puts("\t      FILES AND WRITES THIS PHASE INTO ONE .PHB FILE\n");
  puts("\t      Completed by Gernot Hassenpflug - 16 October 1995\n");
  puts("\n\t      File specs being read in from file SUBPH.CMD");

  /* Open COMMAND file containing file specs */
  if ( (cmdfile = fopen ("SUBPH.CMD", "r")) == NULL )
    { puts(errs[0]); exit_proc(); }

  /* Allocate memory for filenames */
  if ((FNAME = (string *)malloc(2*sizeof(string))) == NULL)
    { puts(errs[9]); exit_proc(); }

  /* Read from SUBPH.CMD and perform the necessary parsing functions */
  /* # = Simulation directive */
  /* % = File definition      */
  /* ! = Comment              */
  do
    { test = getc(cmdfile);

      switch (test)
            { case '!' : do test = getc(cmdfile);
                  while ((test!='\n') && (test!=EOF));
                  line_count++; break;
              case ' ' :
              case '\t':
              case EOF : break;
```

```
            }
    }
  while (test!=EOF);

  /* Close the command file */
  if (fclose(cmdfile) == EOF) exit_proc();

  /* Print success message */
  puts("\n\n Command file processed - no problems encountered.");

  /* Now do the calculations */
  INTINIT(); store();

  /* Exit */
  puts("\n Processing completed - press ENTER...\a");
  scanf("%c",&test);
  exit_proc();
  return 0;
}


/*************************************************************************/
/* This function defines all the error and warning messages.             */
/* Inputs  : None                                                        */
/* Outputs : None                                                        */
/*************************************************************************/

void error_initialize( void )
{
  strcpy(errs[0],  "\nERROR: Command File \"SUBPH.CMD\" not found in current pat
  => h.\a\nProgram Terminated");
  strcpy(errs[3],  "\nERROR: Unable to parse directive name in line %d of comman
  => d file.\a\nProgram Terminated");
  strcpy(errs[5],  "\nERROR: Unable to parse parameter for directive #%s in line
  =>  %d of\ncommand file.\a\nProgram Terminated");
  strcpy(errs[6],  "\nERROR: Illegal directive #%s in line %d of command file.\a
  => \nProgram Terminated");
  strcpy(errs[9],  "\nERROR: Unable to allocate memory.\a\nProgram Terminated");
  strcpy(errs[13], "\nERROR: Unable to parse variable name in line %d of command
  =>  file.\a\nProgram Terminated");
  strcpy(errs[14], "\nERROR: Variable assignment operator '>' not found in line
  => %d of command file.\a\nProgram Terminated");
  strcpy(errs[15], "\nERROR: Unable to parse value to be assigned to variable %s
  =>  in\nline %d of command file.\a\nProgram Terminated");
  strcpy(errs[16], "\nERROR: Illegal variable %s in line %d of command file.\a\n
  => Program Terminated");
  strcpy(errs[18], "\nERROR: Unable to open data file \"%s\".\a\nProgram Termina
  => ted");
  strcpy(errs[19], "\nERROR: No output file name has been specified in command f
  => ile. Use #OUTNAME\ndirective.\a\nProgram Terminated");
  strcpy(errs[20], "\nERROR: The file prefix \"%s\" in line %d of command file i
  => s longer than\n8 characters.\a\nProgram Terminated");
  strcpy(errs[21], "\nERROR: The directive #ASCFILE in line %d of command file h
  => as no parameter.\a\nProgram Terminated");
  strcpy(wrns[1],  "\nWARNING: The variable %s in line %d has already been defin
  => ed earlier\nin the command file.");
}


/*************************************************************************/
/* This function de-allocates all memory and exits the program.          */
/* Inputs  : None                                                        */
/* Outputs : None                                                        */
/*************************************************************************/

void exit_proc( void )
{ free(Q_VALS);
  free(I_VALS);
  free(FNAME);
  free(P_VALS);
  free(phasevals);
  exit(1);
}

/*************************************************************************/
/* This function parses a Simulation Directive i.e. a line beginning     */
/* with the # character.                                                 */
/* Inputs  : None                                                        */
/* Outputs : None                                                        */
/*************************************************************************/

void parse_directive( void )

{ char name[256], name2[256];
  int  val_test, tst;

  /* Read directive name from file */
  val_test = fscanf(cmdfile, "%s", name);
  if ( (val_test == 0) || (val_test == EOF) )
    { printf(errs[3],line_count); exit_proc(); }
  val_test = 0;

  /* Test for ASCFILE */
  if (strncmp(name,"ASCFILE",7) == 0)
    { do
      tst = getc(cmdfile);
      while ( (tst != 'Y') && (tst != 'N') && (tst != '\n') && (tst != '!') );
      if ( (tst == '\n') || (tst == '!') )
         { printf(errs[21],line_count); exit_proc(); }
      if (tst == 'Y') ascii_file = 1;
      val_test = 1;
    }

  /* Test for OUTNAME */
  if (strncmp(name,"OUTNAME",7) == 0)
    { val_test = fscanf(cmdfile, "%s", name);
      if ( (val_test == 0) || (val_test == EOF) )
        { printf(errs[19],"OUTNAME",line_count); exit_proc(); }
      val_test = 0;
      if (strlen(name)>8) { printf(errs[20],name,line_count); exit_proc(); }

      /* Open data files */
      strcpy(out_name,name);
      strcpy(name2,name);
      strcat(name,".PHB");
      if ( (datfile = fopen(name,"wb")) == NULL )
        { printf(errs[18], name); exit_proc(); }
      else { printf (" Success in opening output file.\n"); }

      strcat(name2,".PB2");
      if ( (datfile2 = fopen(name2,"wb")) == NULL )
        { printf(errs[18], name2); exit_proc(); }
      else { printf (" Success in opening output file.\n"); }
      valid_run = 1;
      val_test = 1;
    }

  /* Test for UPRATE */
  if (strncmp(name,"UPRATE",6) == 0)
    { val_test = fscanf(cmdfile, "%d", &update_rate);
      if ( (val_test == 0) || (val_test == EOF) )
         { printf(errs[5],"UPRATE",line_count); exit_proc(); }
```

```
      val_test = 1;
    }

  /* No valid directive name found */
  if (!val_test) { printf(errs[6],name,line_count); exit_proc(); }
}


/*************************************************************************/
/* This function parses the variable definitions from the command file,  */
/* and assigns their values.                                             */
/* Inputs  : None                                                        */
/* Outputs : None                                                        */
/*************************************************************************/

void parse_variable( void )

{ char  name[256];
  int   val_test;
  float val;

  /* Restore the first character of the variable name to the file */
  ungetc(test,cmdfile);

  /* Read variable name from file */
  val_test = fscanf(cmdfile, "%s", name);
  if ( (val_test == 0) || (val_test == EOF) )
    { printf(errs[13],line_count); exit_proc(); }

  /* Now find the assignment operator '>' */
  while ( (test=getc(cmdfile)) == ' ');
  if (test != '>') { printf(errs[14],line_count); exit_proc(); }

  /* Now read the value to be assigned */
  if ( strncmp(name,"INFILE",6) == 0)
     { printf("\n Filename scanned..."); }
  else
     {val_test = fscanf(cmdfile, "%f", &val);
      if ( (val_test == 0) || (val_test == EOF) )
         { printf(errs[15],line_count); exit_proc(); }
     }

  /* Identify the variable names. This is done according to:
     INTINIT is the first bit (bit zero) of function_log, and thus
     all variables used by INTINIT will be logged in variable_log[0]. */

  /* This is detailed in the following diagrams: */

  /* For function 0: INTINIT
     msb ..............................  lsb  VAR. NAME, BIT, OPTIONAL?
                                      ^^^^^^^^
                                      |||||||
                                      ||||||------ NPRIS      0
                                      |||||-------- R_OFF      1
                                      ||||--------- A_OFF      2
                                      |||---------- BINS       3
                                      ||----------- S_R_BIN    4
                                      |------------ S_A_BIN    5
                                      ------------- N_R_BINS   6
                                      ------------- N_A_BINS   7          */

  val_test = 0;

  /* Test for NPRIS */
  if (strncmp(name,"NPRIS",5) == 0)
    { NPRIS = (int) val; val_test = 1;
      if (variable_log[0] & 1) printf(wrns[1],name,line_count);
      else variable_log[0] += 1;
    }

  /* Test for BINS */
  if (strncmp(name,"BINS",4) == 0)
    { BINS = val; val_test = 1;
      if (variable_log[0] & 8) printf(wrns[1],name,line_count);
      else variable_log[0] += 8;
    }

  /* Test for R_OFF */
  if (strncmp(name, "R_OFF",5) == 0)
     { R_OFF = val; val_test = 1;
       if (variable_log[0] & 2) printf(wrns[1],name,line_count);
       else variable_log[0] += 2;
     }

  /* Test for A_OFF */
  if (strncmp(name, "A_OFF",5) == 0)
     { A_OFF = val; val_test = 1;
       if (variable_log[0] & 4) printf(wrns[1],name,line_count);
       else variable_log[0] += 4;
     }

  /* Test for S_R_BIN */
  if (strncmp(name, "S_R_BIN",7) == 0)
     { S_R_BIN = val; val_test = 1;
       if (variable_log[0] & 16) printf(wrns[1],name,line_count);
       else variable_log[0] += 16;
     }

  /* Test for S_A_BIN */
  if (strncmp(name, "S_A_BIN",7) == 0)
     { S_A_BIN = val; val_test = 1;
       if (variable_log[0] & 32) printf(wrns[1],name,line_count);
       else variable_log[0] += 32;
     }

  /* Test for N_R_BINS */
  if (strncmp(name, "N_R_BINS",8) == 0)
     { N_R_BINS = val; val_test = 1;
       if (variable_log[0] & 64) printf(wrns[1],name,line_count);
       else variable_log[0] += 64;
     }

  /* Test for N_A_BINS */
  if (strncmp(name, "N_A_BINS",8) == 0)
     { N_A_BINS = val; val_test = 1;
       if (variable_log[0] & 128) printf(wrns[1],name,line_count);
       else variable_log[0] += 128;
     }

  /* Test for INFILE */
  if (strncmp(name,"INFILE",6) == 0)
    { val_test = fscanf(cmdfile,"%s",name);
      if ( (val_test == 0) || (val_test == EOF) )
      { printf(errs[5],"INFILE",line_count);exit_proc(); }
      val_test = 0;
      if (strlen(name) > 8) { printf(errs[20],name,line_count); exit_proc(); }

      /* write name into FNAME array */
      strcpy(FNAME[fil_index],name);
      printf("\t Written to FNAME : %s",name);
      val_test = 1;
      valid_run = 1;
```

```
    }

  /* No valid variable found */
  if (!val_test) { printf(errs[16],name,line_count); exit_proc(); }
}


/*************************************************************************/
/* This function calculates the phase differences between 2 files.       */
/* Inputs  : None                                                        */
/* Outputs : None                                                        */
/*************************************************************************/

void INTINIT( void )
{
  char               name[256];
  unsigned long int MAX, Foff, Moff;
  int                fil_loop,r;
  long int           temp_Q;
  unsigned char      val_bin;

  /* initialize values for loops */
  MAX = (unsigned long int)N_R_BINS*(unsigned long int)N_A_BINS;
  Foff = (unsigned long int)(S_A_BIN)*(unsigned long int)(BINS)+(unsigned long i
  => nt)(S_R_BIN);
  Moff = (unsigned long int)(S_A_BIN)*(unsigned long int)(BINS)+(unsigned long i
  => nt)(S_R_BIN + R_OFF);

  /* Creating Array Q_VALS */
  if ( (Q_VALS = (unsigned char *)calloc(MAX,sizeof(unsigned char))) == NULL )
     { puts(errs[9]); exit_proc(); }

  /* Creating Array phasevals */
  if ( (phasevals = (unsigned char *)calloc(MAX,sizeof(unsigned char))) == NULL
  => )
     { puts(errs[9]); exit_proc(); }

  /* Creating Array I_VALS */
  if ( (I_VALS = (unsigned char *)calloc(MAX,sizeof(unsigned char))) == NULL )
     { puts(errs[9]); exit_proc(); }

  /* Creating Array P_VALS */
  if ( (P_VALS = (float *)calloc(MAX,sizeof(float))) == NULL )
     { puts(errs[9]); exit_proc(); }

  /* read input file names stored in array FNAME, writing data into arrays */
  /* Q_VALS and I_VALS so there are (NPRIS times the number of range bins) */
  /* elements in each array */
  fil_loop = 0; q_index = 0;

  for (fil_loop = 0; fil_loop < (fil_index+1); fil_loop++)
     { strcpy(name,FNAME[fil_loop]);
       printf("\n Reading from : %s",FNAME[fil_loop]); /* first file */
       if ( (indatfile = fopen(name,"rb")) == NULL )
         { printf(errs[18],name); exit_proc(); }
       printf("\t File opened OK... ");

       if (fil_loop == 0)
         { q_index = 0; printf("Reading from first file\n");
           Shift(Foff);
           for (r=0;r<N_A_BINS;r++)
             {
                Read(N_R_BINS);
                Shift(BINS - N_R_BINS);
             }
         }
       else
         { printf("Reading from second file\n");
           q_index=0;
           Shift(Moff);
           for (r=0;r<N_A_BINS;r++)
            {
              Readsub(N_R_BINS);
              Shift(BINS - N_R_BINS);
            }
         }
     }
  printf("\n File reading and Q calculations done\n");
}

/*************************************************************************/
/* This function calculates a shift into the image file                  */
/* Inputs  : Offset                                                      */
/* Outputs : None                                                        */
/*************************************************************************/

void Shift( unsigned long int maxindex )

{ unsigned long int index;
  unsigned char     val_bin;

  for (index=0;index<maxindex;index++)
    { val_bin = getc(indatfile);  /* I value read */
      val_bin = getc(indatfile);  /* Q value read */
    }
}

/*************************************************************************/
/* This function reads a specified number of azimuth bins                */
/* Inputs  : Bin number                                                  */
/* Outputs : Writes to Q_VALS                                            */
/*************************************************************************/

void Read( int binnumber )

{ int           index;
  unsigned char val_bin;

  for (index=0;index<binnumber;index++)
    { val_bin = getc(indatfile);  /* I value read */
      I_VALS[q_index] = (unsigned char)val_bin;

      val_bin = getc(indatfile);  /* Q value read */
      Q_VALS[q_index++] = (unsigned char)val_bin;
    }
}

/*************************************************************************/
/* This function subtracts two bin ranges from one another               */
/* Inputs  : Number of bins to subtract                                  */
/* Outputs : Writes to Q_VALS                                            */
/*************************************************************************/

void Readsub( int binnumber )

{ int      index;
  float         temp_I1, temp_I2, temp_Q1, temp_Q2;
  float         temp_P1, temp_P2;
  float         P1, P2;
  unsigned char val_bin;

  for (index=0;index<binnumber;index++)
```

```
    {
      val_bin = getc(indatfile);  /* I value read */
      temp_I2 = (float)((((int)(val_bin))-127)*2/3);
      temp_I1 = (float)((I_VALS[q_index]-127)*2/3);

      val_bin = getc(indatfile);  /* Q value read */
      temp_Q2 = (float)((((int)(val_bin))-127)*2/3);
      temp_Q1 = (float)((Q_VALS[q_index]-127)*2/3);

      if ((temp_I1 <= 8.0) && (temp_Q1 <= 8.0))
         { temp_I1 = 0.0;
           temp_Q1 = 0.0;
         }
      if ((temp_I2 <= 8.0) && (temp_Q2 <= 8.0))
         { temp_I2 = 0.0;
           temp_Q2 = 0.0;
         }

      if (temp_I1 != 0.0)
       {
        temp_P1 = (float)(atan(temp_Q1/temp_I1));
        P1 = Angle(temp_P1, temp_Q1);
       }
      else if (temp_I1 == 0.0)
       {
        if (temp_Q1 > 0.0)
          P1 = PI/2;
        else if (temp_Q1 < 0.0)
          P1 = PI*3/2;
        else if (temp_Q1 == 0.0)
          P1 = 0.0;
       }
      if (temp_I2 != 0.0)
       {
        temp_P2 = (float)(atan(temp_Q2/temp_I2));
        P2 = Angle(temp_P2, temp_Q2);
       }
      else if (temp_I2 == 0.0)
       {
        if (temp_Q2 > 0.0)
          P2 = PI/2;
        else if (temp_Q2 < 0.0)
          P2 = PI*3/2;
        else if (temp_Q2 == 0.0)
          P2 = 0.0;
       }

      P_VALS[q_index] = ((P2 - P1))/(2*PI)*127.0 + 127.0;
      q_index++;
    }
}

/*************************************************************************/
/* This function calculates the correct angle from the principal angle   */
/* Inputs  : principal angle, Q value                                    */
/* Outputs : angle in range 0 to 2*PI radians                            */
/*************************************************************************/

float Angle( float principal, float Q_Value )

{ float         Total_Angle;
  unsigned char val_bin;

  if (principal >= 0.0)
    { if (Q_Value >= 0.0)
        Total_Angle = principal;
      else
        Total_Angle = PI + principal;
    }
  else if (principal < 0.0)
    {
     if (Q_Value >= 0.0)
       Total_Angle = PI + principal;
     else
       Total_Angle = 2*PI + principal;
    }
  return (Total_Angle);
}

/*************************************************************************/
/* This function stores output files.                                    */
/* Inputs  : None                                                        */
/* Outputs : None                                                        */
/*************************************************************************/

void store( void )

{ register int n;
  unsigned long int posn,
                    MAX;
  char ascii[14];
  FILE *ascfile;
  float f, temp_m, temp_n, scale;
  unsigned char b;

  scale = 3;
  MAX = (unsigned long int)(N_R_BINS)*(unsigned long int)(N_A_BINS);

  /* Write values to output files */
  printf("\n Writing values to data file  %s.PHB",out_name);
  fwrite(P_VALS,sizeof(f),MAX,datfile);
  fclose(datfile);

  for (n=0;n<MAX;n++)
      {
        temp_m = P_VALS[n];
        phasevals[n] = (unsigned char)temp_m;
      }
  fwrite(phasevals,sizeof(b),MAX,datfile2); /* phase values converted to 0-255 *
  => /
  fclose(datfile2);

  /* Now write the data to an ASCII file, if required */
  if (ascii_file)
    { /* Make up the file name */
      strcpy(ascii,out_name);
      strcat(ascii,".PHA");
      if ( (ascfile = fopen(ascii,"wt")) == NULL ) printf(errs[18],ascii);

      /* Print the headers */
      printf("\n Writing to ASCII file        %s\n",ascii);
      fprintf(ascfile,"Phase values\n\n");

      /* And print the data */
      for (posn=0; posn<MAX; posn++)
         { fprintf(ascfile, "%f\t", P_VALS[posn]); }
      fclose(ascfile);
    }
 }

/*************************************************************************/
/*                    End of Program Listing                             */
```

```
/*************************************************************************/
```

```
/*************************************************************************/
/* Program to unwrap the interferometric phase of an image file and      */
/* return the height of each data point                                  */
/* --------------------------------------------------------------------- */
/* Original Author     :  Gernot Hassenpflug                             */
/* Date                :  11 November 1995                               */
/* Modified            :  25 September 1996                              */
/* Program listing     :  GEOM.C                                         */
/* Set up requirements :  GEOM.CMD, one Q data file                      */
/* Output destination  :  <filename>.HTB, .HTA, .UPB, .UPA               */
/*************************************************************************/

/* Include library functions */

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

/* Definitions */

#define PI 3.141592654
#define C  3e+8

/* Type definitions */

typedef unsigned char string[14]; /* string definition for filenames    */

/* Declare file pointers */

FILE   *cmdfile,          /* The command file "GEOM.CMD"                 */
       *datfile1,         /* The data file for phase output              */
       *datfile2,         /* The data file for height output             */
       *indatfile;        /* The data file for input                     */

/* Variables assigned values from command file  */

int    NPRIS,             /* No. of pulse repetition intervals    [none] */
       BINS,              /* Number of range bins                 [none] */
       S_R_BIN,S_A_BIN,   /* Starting bins in range and azimuth          */
       N_A_BINS,N_R_BINS,/* Number of bins to read                       */
       PASSES;            /* Flag for Dual Pass or Single Pass           */

float  A2D,               /* A2D Frequency                        [GHz]  */
       V,                 /* Aircraft speed                       [m/s]  */
       TD,                /* time delay to 1st range bin          [secs] */
       BL,                /* Baseline length                      [m]    */
       ALPHA,             /* angle of BL to horizontal            [deg]  */
       HGT,               /* Aircraft Altitude                    [m]    */
       LAMBDA;            /* Radar wavelength                     [m]    */

string *FNAME;            /* pointer to array of input filenames         */

/* Variables assigned values within program */

int    test,              /* Store for character read from command file  */
       line_count,        /* Index to line number of command file        */
       fil_index,         /* Index to the file number being processed    */
       update_rate,       /* Progress report every n-th iteration        */
       valid_run,         /* Indicates whether a file name was defined   */
       ascii_file;        /* Indicates whether an ASCII file is required */

int    variable_log[16], /* Flags indicating variables defined          */
       max_val;           /* Maximum I or Q value in digital form        */
```



```
float *UP_VALS;           /* Array for unwrapped phase values            */
unsigned char *Q_VALS;    /* Array for wrapped phase values              */
unsigned long int q_index;/* Index to the Q value being processed        */
float *H_VALS;            /* Array for height values                     */

char   errs[22][120],     /* Array to hold all error messages            */
       wrns[4][120],      /* Array to hold all warning messages          */
       out_name[14],      /* First part of the file name used for output */
       in_name[14];       /* First part of input file names              */

/* Function Prototypes - in the order that they appear in the code */

  /* Functions concerning command file reading and error checking. */
  void error_initialize( void );
  void exit_proc( void );
  void parse_directive( void );
  void parse_variable( void );

  /* Functions to do with the actual simulation and the log file */
  void FINDHGT( void );
  void store( void );
  void Shift( unsigned long int );
  void Read( int );
  void Readsub( int );
  void Unwrap( int );

/*************************************************************************/
/* Main Program                                                          */
/*************************************************************************/

main()
{
  /* Initialize error messages */
  error_initialize();

  /* Initialize variables and flags */
  line_count = valid_run = ascii_file = 0;
  fil_index = -1; update_rate = 100; max_val = 0;
  for (test=0; test<16; variable_log[test++]=0);

  /* Print headings on the screen */
  /* system("cls");*/
  /* clrscr(); */
  puts("\t THIS PROGRAM FINDS THE HEIGHT OF DATA POINTS IN A SPECIFIED\n");
  puts("\t      FILE AND WRITES THIS INFORMATION INTO ONE .HTB FILE\n");
  puts("\t       Completed by Gernot Hassenpflug - 16 October 1995\n");
  puts("\n File specs being read in from file GEOM.CMD");

  /* Open COMMAND file containing file specs */
  if ( (cmdfile = fopen ("GEOM.CMD", "r")) == NULL )
    { puts(errs[0]); exit_proc(); }

  /* Allocate memory for filenames */
  if ((FNAME = (string *)malloc(sizeof(string))) == NULL)
    { puts(errs[9]); exit_proc(); }

  /* Read from GEOM.CMD and perform the necessary parsing functions */
  /* # = Simulation directive */
  /* % = File definition      */
  /* ! = Comment              */
  do
    { test = getc(cmdfile);

      switch (test)
            { case '!' : do test = getc(cmdfile);
                         while ((test!='\n') && (test!=EOF));
```

```
                line_count++; break;
              case ' ' :
              case '\t':
              case EOF : break;
        case '\n': line_count++; break;
              case '%' : fil_index++; break;
              case '#' : parse_directive(); break;
              default  : parse_variable();
            }
    }
  while (test!=EOF);

  /* Close the command file */
  if (fclose(cmdfile) == EOF) exit_proc();

  /* Print success message */
  puts("\n\n Command file processed - no problems encountered.");

  /* Now do the calculations */
  FINDHGT(); store();

  /* Exit */
  puts("\n Processing completed - press ENTER...\a");
  scanf("%c",&test);
  exit_proc();
  return 0;
}


/*************************************************************************/
/* This function defines all the error and warning messages.             */
/* Inputs  : None                                                        */
/* Outputs : None                                                        */
/*************************************************************************/

void error_initialize( void )
{
  strcpy(errs[0],  "\nERROR: Command File \"GEOM.CMD\" not found in current path
  => .\a\nProgram Terminated");
  strcpy(errs[3],  "\nERROR: Unable to parse directive name in line %d of comman
  => d file.\a\nProgram Terminated");
  strcpy(errs[5],  "\nERROR: Unable to parse parameter for directive #%s in line
  =>  %d of\ncommand file.\a\nProgram Terminated");
  strcpy(errs[6],  "\nERROR: Illegal directive #%s in line %d of command file.\a
  => \nProgram Terminated");
  strcpy(errs[9],  "\nERROR: Unable to allocate memory.\a\nProgram Terminated");
  strcpy(errs[13], "\nERROR: Unable to parse variable name in line %d of command
  =>  file.\a\nProgram Terminated");
  strcpy(errs[14], "\nERROR: Variable assignment operator '>' not found in line
  => %d of command file.\a\nProgram Terminated");
  strcpy(errs[15], "\nERROR: Unable to parse value to be assigned to variable %s
  =>  in\nline %d of command file.\a\nProgram Terminated");
  strcpy(errs[16], "\nERROR: Illegal variable %s in line %d of command file.\a\n
  => Program Terminated");
  strcpy(errs[18], "\nERROR: Unable to open data file \"%s\".\a\nProgram Termina
  => ted");
  strcpy(errs[19], "\nERROR: No output file name has been specified in command f
  => ile. Use #OUTNAME\ndirective.\a\nProgram Terminated");
  strcpy(errs[20], "\nERROR: The file prefix \"%s\" in line %d of command file i
  => s longer than\n8 characters.\a\nProgram Terminated");
  strcpy(errs[21], "\nERROR: The directive #ASCFILE in line %d of command file h
  => as no parameter.\a\nProgram Terminated");
  strcpy(wrns[1],  "\nWARNING: The variable %s in line %d has already been defin
  => ed earlier\nin the command file.");
}
```

```
/*************************************************************************/
/* This function de-allocates all memory and exits the program.          */
/* Inputs  : None                                                        */
/* Outputs : None                                                        */
/*************************************************************************/

void exit_proc( void )
{ free(UP_VALS); free(FNAME); exit(1); }


/*************************************************************************/
/* This function parses a Simulation Directive i.e. a line beginning     */
/* with the # character.                                                 */
/* Inputs  : None                                                        */
/* Outputs : None                                                        */
/*************************************************************************/

void parse_directive( void )

{ char name[256], name2[256];
  int  val_test, tst;

  /* Read directive name from file */
  val_test = fscanf(cmdfile, "%s", name);
  if ( (val_test == 0) || (val_test == EOF) )
    { printf(errs[3],line_count); exit_proc(); }
  val_test = 0;

  /* Test for ASCFILE */
  if (strncmp(name,"ASCFILE",7) == 0)
    { do
      tst = getc(cmdfile);
      while ( (tst != 'Y') && (tst != 'N') && (tst != '\n') && (tst != '!') );
      if ( (tst == '\n') || (tst == '!') )
         { printf(errs[21],line_count); exit_proc(); }
      if (tst == 'Y') ascii_file = 1;
      val_test = 1;
    }

  /* Test for OUTNAME */
  if (strncmp(name,"OUTNAME",7) == 0)
    { val_test = fscanf(cmdfile, "%s", name);
      if ( (val_test == 0) || (val_test == EOF) )
        { printf(errs[19],"OUTNAME",line_count); exit_proc(); }
      val_test = 0;
      if (strlen(name)>8) { printf(errs[20],name,line_count); exit_proc(); }

      /* Open data file1 */
      strcpy(out_name,name);
      strcpy(name2,name);
      strcat(name,".UPB");
      if ( (datfile1 = fopen(name,"wb")) == NULL )
        { printf(errs[18], name); exit_proc(); }
      else { printf (" Success in opening output file.\n"); }

      /* Open data file2 */
      strcat(name2,".HTB");
      if ( (datfile2 = fopen(name2,"wb")) == NULL )
        { printf(errs[18], name2); exit_proc(); }
      else { printf (" Success in opening output file.\n"); }

      valid_run = 1;
      val_test = 1;
    }
```

```
  /* Test for UPRATE */
  if (strncmp(name,"UPRATE",6) == 0)
    { val_test = fscanf(cmdfile, "%d", &update_rate);
      if ( (val_test == 0) || (val_test == EOF) )
          { printf(errs[5],"UPRATE",line_count); exit_proc(); }
      val_test = 1;
    }

  /* No valid directive name found */
  if (!val_test) { printf(errs[6],name,line_count); exit_proc(); }
}


/*************************************************************************/
/* This function parses the variable definitions from the command file,  */
/* and assigns their values.                                             */
/* Inputs  : None                                                        */
/* Outputs : None                                                        */
/*************************************************************************/

void parse_variable( void )

{ char  name[256];
  int   val_test;
  float val;

  /* Restore the first character of the variable name to the file */
  ungetc(test,cmdfile);

  /* Read variable name from file */
  val_test = fscanf(cmdfile, "%s", name);
  if ( (val_test == 0) || (val_test == EOF) )
    { printf(errs[13],line_count); exit_proc(); }

  /* Now find the assignment operator '>' */
  while ( (test=getc(cmdfile)) == ' ');
  if (test != '>') { printf(errs[14],line_count); exit_proc(); }

  /* Now read the value to be assigned */
  if ( strncmp(name,"INFILE",6) == 0)
     { printf("\n Filename scanned..."); }
  else
     {val_test = fscanf(cmdfile, "%f", &val);
      if ( (val_test == 0) || (val_test == EOF) )
         { printf(errs[15],line_count); exit_proc(); }
     }

  /* Identify the variable names. This is done according to:
     FINDHGT is the first bit (bit zero) of function_log, and thus
     all variables used by FINDHGT will be logged in variable_log[0]. */

  /* This is detailed in the following diagrams: */

  /* For function 0: FINDHGT
     msb ..................................... lsb  VAR. NAME, BIT, OPTIONAL?

                          ||||||||||||||
                          |||||||||||||------ NPRIS      0
                          |||||||||||-------- BINS       1
                          ||||||||||--------- S_R_BIN    2
                          |||||||||---------- S_A_BIN    3
                          ||||||||----------- N_R_BINS   4
                          |||||||------------ N_A_BINS   5
                          ||||||------------- PASSES     6
                          |||||-------------- V          7
                          ||||--------------- HGT        8
                          |||---------------- BL         9
                          ||----------------- ALPHA      10
                          |------------------ A2D        11
                          ------------------- TD         12
                           ------------------ LAMBDA     13   */


  val_test = 0;

  /* Test for NPRIS */
  if (strncmp(name,"NPRIS",5) == 0)
    { NPRIS = (int) val; val_test = 1;
      if (variable_log[0] & 1) printf(wrns[1],name,line_count);
      else variable_log[0] += 1;
    }

  /* Test for BINS */
  if (strncmp(name,"BINS",4) == 0)
    { BINS = (int) val; val_test = 1;
      if (variable_log[0] & 2) printf(wrns[1],name,line_count);
      else variable_log[0] += 2;
    }

  /* Test for S_R_BIN */
  if (strncmp(name, "S_R_BIN",7) == 0)
     { S_R_BIN = (int) val; val_test = 1;
       if (variable_log[0] & 4) printf(wrns[1],name,line_count);
       else variable_log[0] += 4;
     }

  /* Test for S_A_BIN */
  if (strncmp(name, "S_A_BIN",7) == 0)
     { S_A_BIN = (int) val; val_test = 1;
       if (variable_log[0] & 8) printf(wrns[1],name,line_count);
       else variable_log[0] += 8;
     }

  /* Test for N_R_BINS */
  if (strncmp(name, "N_R_BINS",8) == 0)
     { N_R_BINS = (int) val; val_test = 1;
       if (variable_log[0] & 16) printf(wrns[1],name,line_count);
       else variable_log[0] += 16;
     }

  /* Test for N_A_BINS */
  if (strncmp(name, "N_A_BINS",8) == 0)
     { N_A_BINS = (int) val; val_test = 1;
       if (variable_log[0] & 32) printf(wrns[1],name,line_count);
       else variable_log[0] += 32;
     }

  /* Test for PASSES */
  if (strncmp(name, "PASSES",6) == 0)
     { PASSES = (int) val; val_test = 1;
       if (variable_log[0] & 64) printf(wrns[1],name,line_count);
       else variable_log[0] += 64;
     }

  /* Test for V */
  if (strncmp(name, "V",1) == 0)
     { V = val; val_test = 1;
       if (variable_log[0] & 128) printf(wrns[1],name,line_count);
       else variable_log[0] += 128;
     }

  /* Test for HGT */
```

```
  if (strncmp(name, "HGT",3) == 0)
     { HGT = val; val_test = 1;
       if (variable_log[0] & 256) printf(wrns[1],name,line_count);
       else variable_log[0] += 256;
     }

  /* Test for BL */
  if (strncmp(name, "BL",2) == 0)
     { BL = val; val_test = 1;
       if (variable_log[0] & 512) printf(wrns[1],name,line_count);
       else variable_log[0] += 512;
     }

  /* Test for ALPHA */
  if (strncmp(name, "ALPHA",5) == 0)
     { ALPHA = val; val_test = 1;
       if (variable_log[0] & 1024) printf(wrns[1],name,line_count);
       else variable_log[0] += 1024;
     }

  /* Test for A2D */
  if (strncmp(name, "A2D",3) == 0)
     { A2D = val; val_test = 1;
       if (variable_log[0] & 2048) printf(wrns[1],name,line_count);
       else variable_log[0] += 2048;
     }

  /* Test for TD */
  if (strncmp(name, "TD",2) == 0)
     { TD = val; val_test = 1;
       if (variable_log[0] & 4096) printf(wrns[1],name,line_count);
       else variable_log[0] += 4096;
     }

  /* Test for LAMBDA */
  if (strncmp(name, "LAMBDA",6) == 0)
     { LAMBDA = val; val_test = 1;
       if (variable_log[0] & 8192) printf(wrns[1],name,line_count);
       else variable_log[0] += 8192;
     }

  /* Test for INFILE */
  if (strncmp(name,"INFILE",6) == 0)
    { val_test = fscanf(cmdfile,"%s",name);
      if ( (val_test == 0) || (val_test == EOF) )
      { printf(errs[5],"INFILE",line_count);exit_proc(); }
      val_test = 0;
      if (strlen(name) > 8) { printf(errs[20],name,line_count); exit_proc(); }

      /* write name into FNAME array */
      strcpy(FNAME[fil_index],name);
      printf("\t Written to FNAME : %s",name);
      val_test = 1;
      valid_run = 1;
    }

  /* No valid variable found */
  if (!val_test) { printf(errs[16],name,line_count); exit_proc(); }
}


/*************************************************************************/
/* This function unwraps the phase of a phase difference file. Then the  */
/* unwrapped phase is converted to a height.                             */
/* Inputs  : None                                                        */
/* Outputs : None                                                        */
```

```
/*************************************************************************/

void FINDHGT( void )
{
  char              name[256];
  unsigned long int MAX, Moff, Foff;
  int               r_index, n;
  unsigned char     val_bin;
  float             subterm, temp_p, LookAng, range, range2,
                    delta_p, offset_p, offset_r, TperBIN, t, s, q;
  double            a, b;
  float             f;

  /* initialize values for loops */
  MAX = (unsigned long int)N_R_BINS*(unsigned long int)N_A_BINS;
  Foff = (unsigned long int)(S_R_BIN)*(unsigned long int)(NPRIS)+(unsigned long
  => int)(S_A_BIN);

  /* Creating Array for storing Unwrapped phase values */
  if ( (UP_VALS = (float *)calloc(MAX,sizeof(float))) == NULL )
     { puts(errs[9]); exit_proc(); }

  /* Creating Array for storing Height values */
  if ( (H_VALS = (float *)calloc(MAX,sizeof(float))) == NULL )
     { puts(errs[9]); exit_proc(); }

  /* Creating Array for storing wrapped phase values */
  if ( (Q_VALS = (unsigned char *)calloc(MAX,sizeof(unsigned char))) == NULL )
     { puts(errs[9]); exit_proc(); }

  /* read file name stored in array FNAME, writing data into array */
  /* UP_VALS[]                                                     */
  q_index = 0; r_index = 0;

  strcpy(name,FNAME[fil_index]);
  printf("\n Reading from : %s",FNAME[fil_index]);
  if ( (indatfile = fopen(name,"rb")) == NULL )
     { printf(errs[18],name); exit_proc(); }
       printf("\t File opened OK...");
       q_index = 0; printf(" Reading from File \n");

/* ------------ Phase unwrapping algorithm - Noise-free version ------------ */

      fread(Q_VALS,sizeof(val_bin),MAX,indatfile);
      q_index = 0;
      for (r_index=0;r_index<N_A_BINS;r_index++)
        { Unwrap(N_R_BINS); }
     printf(" Unwrapping completed\n");

/* ---------- Height Conversion Algorithm ------------- */

/* see diagrams in thesis for graphic layout of geometry used in
   deriving equations for this phase to height transformation */

printf(" Starting calculations\n");
        q_index = 0; r_index = 0;
        subterm = LAMBDA/(2*PASSES*PI*BL);  /* radians */
        printf (" Subterm is: %f\n",subterm);
        offset_r = TD*C/2;
        for (r_index=0;r_index<N_A_BINS;r_index++)
           for (n=0;n<N_R_BINS;n++)
             {
               TperBIN = 1/A2D;
               range = offset_r + (((float)(n)*TperBIN))*C/2;
               t = range*range;
               s = HGT*HGT;
```

```
                q = t - s;
                a = sqrt( (double)(q) );
                b = a + (double)(BL);
                range2 = (float)(sqrt( (double)(((float)(b*b)) + HGT*HGT) ) );
                temp_p = (range2 - range)*2*PI*PASSES/LAMBDA;
                LookAng = asin(subterm*temp_p);
                H_VALS[q_index] = range*cos(LookAng);
                q_index++;
              }

  printf("\n File reading and height calculations done\n");
}

/*************************************************************************/
/* This function unwraps the phase of one row of range bins              */
/* Inputs  : Bin number                                                  */
/* Outputs : Writes to UP_VALS                                           */
/*************************************************************************/

void Unwrap( int binnumber )

{ int           index;
  unsigned char val_bin;
  float         temp_f, temp_f2;

  for (index=0;index<binnumber;index++)
    { if (index == 0)
       { UP_VALS[q_index] = (float)(Q_VALS[q_index]+127); }
      else
        { UP_VALS[q_index] = Q_VALS[q_index]+127 + UP_VALS[q_index-1]; }
      q_index++;
    }
}

/*************************************************************************/
/* This function stores output files.                                    */
/* Inputs  : None                                                        */
/* Outputs : None                                                        */
/*************************************************************************/

void store( void )

{ register int n;
  unsigned long int posn,
                    MAX;
  char ascii[14];
  FILE *ascfile;
  float f;

  MAX = (unsigned long int)(N_R_BINS)*(unsigned long int)(N_A_BINS);

  /* Write values to output files */
  printf("\n Writing unwrapped phase values to data file  %s.UPB",out_name);
  fwrite(UP_VALS,sizeof(float),MAX,datfile1);
  fclose(datfile1);

  /* Write values to output files */
  printf("\n Writing height values to data file  %s.HTB",out_name);
  fwrite(H_VALS, sizeof(f),MAX,datfile2);
  fclose(datfile2);

  /* Now write the data to an ASCII file, if required */
  if (ascii_file)
    { /* Make up the file name */
      strcpy(ascii,out_name);
      strcat(ascii,".UPA");
      if ( (ascfile = fopen(ascii,"wt")) == NULL ) printf(errs[18],ascii);

      /* Print the headers */
      printf("\n\n Writing to ASCII file        %s",ascii);
      fprintf(ascfile,"Unwrapped Phase values\n\n");

      /* And print the data */
      for (posn=0; posn<MAX; posn++)
         { fprintf(ascfile, "%10f3\t",(double)UP_VALS[posn]); }
      fclose(ascfile);
    }

  /* Now write the data to an ASCII file, if required */
  if (ascii_file)
    { /* Make up the file name */
      strcpy(ascii,out_name);
      strcat(ascii,".HTA");
      if ( (ascfile = fopen(ascii,"wt")) == NULL ) printf(errs[18],ascii);

      /* Print the headers */
      printf("\n Writing to ASCII file        %s\n",ascii);
      fprintf(ascfile,"Height values\n\n");

      /* And print the data */
      for (posn=0; posn<MAX; posn++)
         { fprintf(ascfile, "%10f3\t",(double)H_VALS[posn]); }
      fclose(ascfile);
    }
}

/*************************************************************************/
/*                      End of Program Listing                           */
/*************************************************************************/
```

Appendix E

MathCAD Worksheets for the Analysis of the SASAR VHF SAR System

The following MathCAD worksheets are included in this Appendix. For further information as to their applicability, see Chapter 4.

1. **Worksheet A**: This MathCAD worksheet simulates the effect of phase decorrelation on the phase error, using the method outlined by Li and Goldstein.

2. **Worksheet B**: This MathCAD worksheet calculates the optimum decorrelation, balancing baseline decorrelation with height error sensitivity to show the effects of these two processes on height error. The possible baselines for a given InSAR system are then calculated and used to calculate fringe spacing and maximum imagable slopes for an unconstrained InSAR system. The effects of layover and shadowing are shown and the range of slopes imagable by a given InSAR system is determined.

3. **Worksheet C**: This MathCAD worksheet contains a parameter sensitivity spreadsheet for an InSAR system. The effect on height error of each InSAR parameter is quantified for a given system. The sensitivity equations are derived from the equations for the radar-target geometry.

4. **Worksheet D**: This MathCAD worksheet examines the effect of incidence angle on the SASAR performance as an InSAR platform. The re-

lationship between incidence angle and radar altitude, and the baseline constraints owing to fringe sampling requirements are presented. The effect of terrain slope on SASAR performance is described by means of sets of curves.

WORKSHEET A

1. Effect of Decorrelation on the rms Phase Error in InSAR

Li and Goldstein [Studies of Multibaseline spaceborne interferometric SARs] outline a model for simulating the effects of decorrelation on phase measurements. The decorrelation is a combination of the thermal noise and the geometric decorrelation caused by the InSAR baseline.

The model has been implemented below. The results allow an estimate of phase estimation error to be made for a given decorrelation and number of interferometric looks.

Two images are created, consisting of two gaussian random vectors P1 and P2. The variance of the vectors is determined by the decorrelation. The interferometric phase of each pixel is estimated and the standard deviation of the interferogram is expressed in degrees and plotted against the decorrelation.

Each vector P1 and P2 has 1000 elements: $M := 1000$ $i := 0..M - 1$

P1 consists of vectors a and b
P2 consists of vectors a and c

The decorrelation varies from 0.1 to 0.9: $d := 1, 2..9$

Correlation coefficient gamma: $g_d := 1 - \frac{d}{10}$

The variance of vector a is equal to the correlation coefficient, while the variance of vectors b and c is equal to the decorrelation. The variances and standard deviations of the real and imaginary parts of each of the vectors a (a1 and a2), b (b1 and b2) and c (c1 and c2) has been calculated and is expressed below:

$d2_d := \frac{d}{20}$ $d2r_d := \sqrt{d2_d}$ $d2r_4 = 0.447$

$g2_d := \frac{g_d}{2}$ $g2r_d := \sqrt{g2_d}$ $g2r_4 = 0.548$

A test was made for a decorrelation of 0.4

The vector components are created using a Gaussian random number generator:

$a1_d := rnorm(M, 0, g2r_d)$ $a2_d := rnorm(M, 0, g2r_d)$

$b1_d := rnorm(M, 0, d2r_d)$ $b2_d := rnorm(M, 0, d2r_d)$

$c1_d := rnorm(M, 0, d2r_d)$ $c2_d := rnorm(M, 0, d2r_d)$

The vectors are created:

$a_d := a1_d + a2_d \cdot \sqrt{-1}$ $b_d := b1_d + b2_d \cdot \sqrt{-1}$ $c_d := c1_d + c2_d \cdot \sqrt{-1}$

The image point vectors are created, and the interferogram formed:

$P1_d := a_d + b_d$ $P2_d := a_d + c_d$ $R1_{i,d} := (P1_d)_i \cdot \overline{(P2_d)_i}$

1-Look Simulation:

The image phase and standard deviation is calculated:

$$RP_{i,d} := \arg\left(R1_{i,d}\right) \qquad ph\_err_d := \frac{stdev\left(RP^{<d>}\right)}{2\cdot\pi} \qquad deg\_err_d := ph\_err_d \cdot 360$$

2-Look Simulation:

The image phase and standard deviation is calculated: $k := 0 .. M - 2$

$$Look2_{k,d} := \frac{1}{2}\cdot\sum_{h=0}^{1} R1_{k+h,d} \qquad RP2_{k,d} := \arg\left(Look2_{k,d}\right) \qquad ph\_2err_d := \frac{stdev\left(RP2^{<d>}\right)}{2\cdot\pi}$$

$$deg\_2err_d := ph\_2err_d \cdot 360$$

4-Look Simulation:

The image phase and standard deviation is calculated: $l := 0 .. M - 4$

$$Look4_{l,d} := \frac{1}{4}\cdot\sum_{h=0}^{3} R1_{l+h,d} \qquad RP4_{l,d} := \arg\left(Look4_{l,d}\right) \qquad ph\_4err_d := \frac{stdev\left(RP4^{<d>}\right)}{2\cdot\pi}$$

$$deg\_4err_d := ph\_4err_d \cdot 360$$

8-Look Simulation:

The image phase and standard deviation is calculated: $q := 0 .. M - 8$

$$Look8_{q,d} := \frac{1}{8}\cdot\sum_{h=0}^{7} R1_{q+h,d} \qquad RP8_{q,d} := \arg\left(Look8_{q,d}\right) \qquad ph\_8err_d := \frac{stdev\left(RP8^{<d>}\right)}{2\cdot\pi}$$

$$deg\_8err_d := ph\_8err_d \cdot 360$$

16-Look Simulation:

The image phase and standard deviation is calculated: $r := 0 .. M - 16$

$$Look16_{r,d} := \frac{1}{16}\cdot\sum_{h=0}^{15} R1_{r+h,d} \qquad RP16_{r,d} := \arg\left(Look16_{r,d}\right) \qquad ph\_16err_d := \frac{stdev\left(RP16^{<d>}\right)}{2\cdot\pi}$$

$$deg\_16err_d := ph\_16err_d \cdot 360$$

32-Look Simulation:

The image phase and standard deviation is calculated: $s := 0 .. M - 32$

$$Look32_{s,d} := \frac{1}{32}\cdot\sum_{h=0}^{31} R1_{s+h,d} \qquad RP32_{s,d} := \arg\left(Look32_{s,d}\right) \qquad ph\_32err_d := \frac{stdev\left(RP32^{<d>}\right)}{2\cdot\pi}$$

$$deg\_32err_d := ph\_32err_d \cdot 360$$

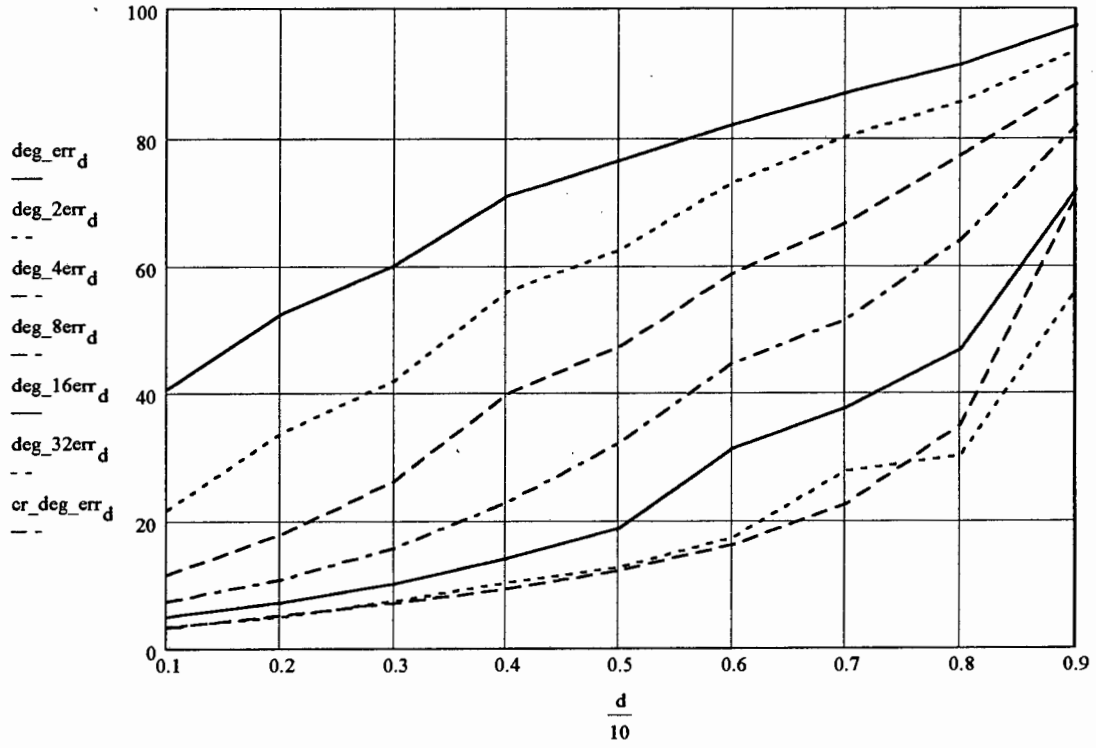
The rms phase errors in tabular form, as a function of looks (columns) and decorrelation (rows):

cycles ph_err_d	deg deg_err_d	deg deg_2err_d	deg deg_4err_d	deg deg_8err_d	deg deg_16err_d	deg deg_32err_d
0.113	40.843	21.888	11.774	7.61	5.248	3.65
0.146	52.739	33.893	18.115	11.028	7.362	5.231
0.167	60.274	42.057	26.362	15.887	10.33	7.593
0.198	71.224	56.217	39.953	23.136	14.3	10.516
0.213	76.841	62.728	47.441	32.323	19.044	12.896
0.229	82.312	73.259	59.02	44.808	31.372	17.516
0.242	87.201	80.481	66.939	51.838	37.815	27.997
0.254	91.516	85.804	77.643	64.328	46.983	30.334
0.271	97.464	93.731	88.544	82.236	72.207	56.423

The Cramer-Rao bound is a Maximum Likelihood Estimator for rms phase estimation error. The phase error appproaches the Cramer-Rao bounding value for a given number of looks as the number of looks increases, but is inaccurate for large decorrelation values, as can be seen by the graph.

Cramer Rao Bound: for 32 looks: $N_l := 32$

$$cr\_ph\_err_d := \frac{\sqrt{1 - (g_d)^2}}{\sqrt{2 \cdot N_l \cdot g_d}} \qquad cr\_deg\_err_d := cr\_ph\_err_d \cdot \frac{360}{2 \cdot \pi}$$



WORKSHEET B

1. Effect of Decorrelation on Height Error

$SNR_{dB} := \begin{pmatrix} 10 \\ 20 \\ 80 \end{pmatrix}$ Assumed SNR [dB]

$i := 0, 1 .. 2$

$SNR_i := 10^{\frac{SNR_{dB_i}}{10}}$

$N_1 := 16$ Assumed number of looks

$r_1 := 40800$ Assumed slant range [km] for aircraft height of 10.7 km

$Q := 2$ Factor for repeat pass InSAR

$\alpha_1 := \frac{75.8}{360} \cdot 2 \cdot \pi$ Look angle [rad] Assumed at mid swath for the simulation

$\theta_b := \alpha_1$ optimum baseline tilt angle is set to the look angle at mid swath

$\delta_{SR} := 26$ Slant range resolution [m]

$B := 6605$ Critical baseline for assumed geometry, where geometric correlation is zero [m]

$\lambda := 2.13$ VHF wavelength [m]

$K_0 := \frac{Q}{\lambda \cdot r_1 \cdot \sin(\alpha_1)}$ $K_0 = 2.374 \cdot 10^{-5}$

$K_1 := K_0 \cdot \delta_{SR} \cdot \cos(\alpha_1)$ $K_1 = 1.514 \cdot 10^{-4}$ Inverse of the Critical Baseline

$K_2 := \frac{1}{2 \cdot \pi \cdot K_0 \cdot \cos(\alpha_1 - \theta_b)}$ $K_2 = 6.704 \cdot 10^{3}$

$K_{3_i} := \frac{1}{1 + \frac{1}{SNR_i}}$ $K_3 = \begin{pmatrix} 0.909 \\ 0.99 \\ 1 \end{pmatrix}$ Thermal noise degradation

$\alpha := 1 - K_1 \cdot B$ $\gamma_i := K_{3_i} \cdot \alpha$ $\delta := 1 - \gamma$ $\delta = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ Allowable decorrelation is limited by phase unwrapping constraints

$\sigma_\phi := \frac{88.5}{360} \cdot 2 \cdot \pi$ phase error for decorrelation of one, in [rads] read off from simulated curves

$\sigma_z := \frac{K_2}{B} \cdot \sigma_\phi$ $\sigma_z = 1.568$ height error in [m] at decorrelation of one

Plots of the effect of decorrelation on height error for SNR of 10dB, 20dB and 80dB. The decorrelation depends on the baseline length. The model for phase error dependence on decorrelation used is the Cramer Rao bound model. As can be seen, the optimum decorrelation problem does not occur for low SNR. A suitable decorrelation can be chosen at will.

$d := 1, 2 .. 99$ $g_d := 1 - \frac{d}{100}$

$s_d := \frac{1}{\sqrt{2 \cdot N_1}} \cdot \frac{\sqrt{1 - (g_d)^2}}{g_d}$ Phase error dependence on correlation using the Cramer Rao model

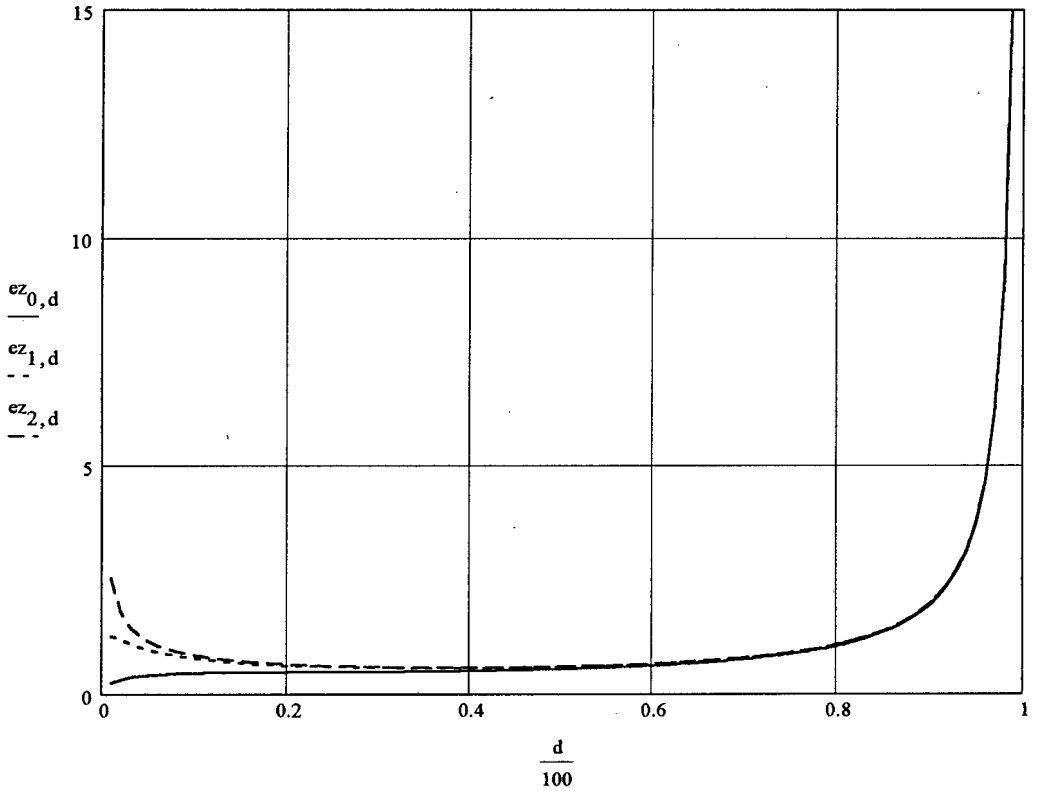
$Bl_{i,d} := \frac{1 - K_{3_i} \cdot g_d}{K_1}$ Baseline length as a function of correlation

$$ez_{i,d} := \frac{K_2}{B1_{i,d}} \cdot s_d$$

Height error as a function of decorrelation

$$ez2_{i,d} := \frac{\frac{\delta_{SR} \cdot \cos(\alpha_1)}{2 \cdot \pi \cdot K3_i \cdot \cos(\alpha_1 - \theta_b)}}{\frac{1}{K3_i} - g_d} \cdot s_d$$

Height error function simplified. The K term in the denominator is the SNR factor, and tends to unity as SNR goes to infinity (effectively above 20dB)



The curves show allow choice of the optimum decorrelation for a particular SNR, number of looks and phase error model.

2. Optimal and Critical Baseline, and Fringe Spacing

For the VHF system with parameters as assumed, the optimum decorrelation is about 0.4. However, any decorrelation chosen between about 0.1 and 0.8 will not affect the height error significantly. From the decorreltaion, the baseline to be used is determined:

$$B_c := \frac{1}{K_1} \qquad B_c = 6.605 \cdot 10^3 \quad [m]$$

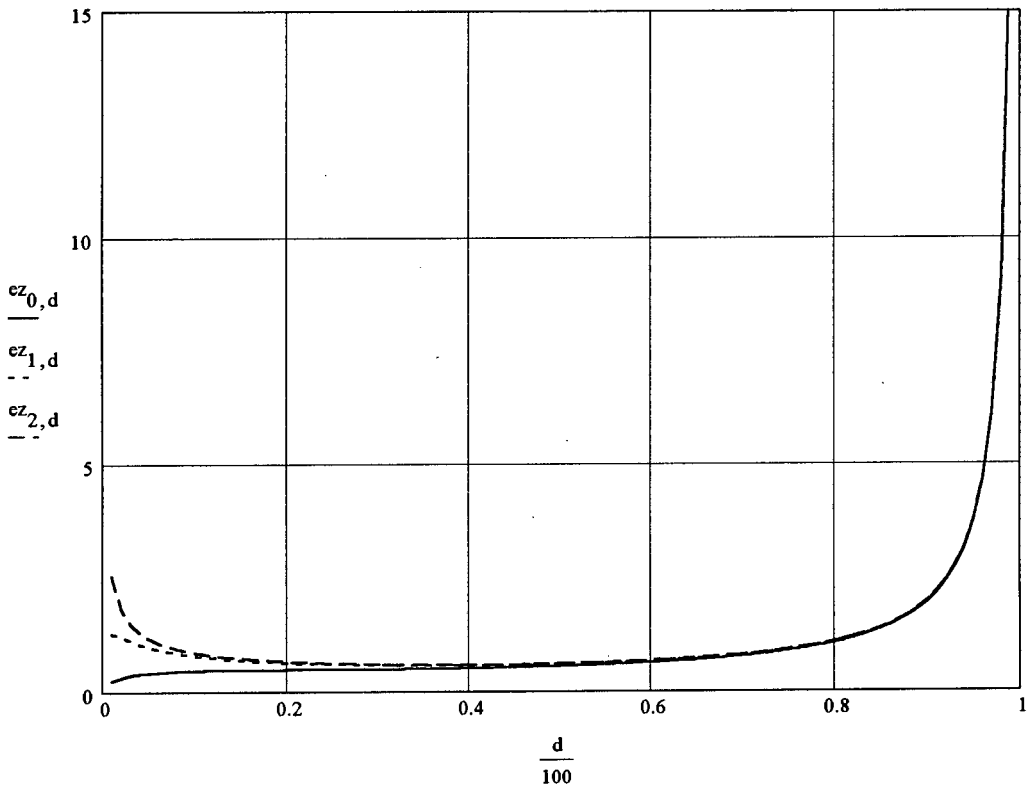
Critical baseline, where decorrelation is unity

Range of acceptabe Baselines, corresponding to decorrelation of 0.2 to 0.6, and depending on the SNR:

$$B_{shortest} := \frac{1 - (1 - 0.1) \cdot K_3}{K_1} \qquad B_{shortest} = \begin{pmatrix} 1.201 \cdot 10^3 \\ 719.317 \\ 660.464 \end{pmatrix}$$

$ez_{i,d} := \frac{K_2}{B1_{i,d}} \cdot s_d$ Height error as a function of decorrelation

$ez2_{i,d} := \frac{\frac{\delta_{SR} \cdot \cos(\alpha_1)}{2 \cdot \pi \cdot K_{3_i} \cdot \cos(\alpha_1 - \theta_b)}}{\frac{1}{K_{3_i}} - g_d} \cdot s_d$ Height error function simplified. The K term in the denominator is the SNR factor, and tends to unity as SNR goes to infinity (effectively above 20dB)



The curves show allow choice of the optimum decorrelation for a particular SNR, number of looks and phase error model.

2. Optimal and Critical Baseline, and Fringe Spacing

For the VHF system with parameters as assumed, the optimum decorrelation is about 0.4. However, any decorrelation chosen between about 0.1 and 0.8 will not affect the height error significantly. From the decorreltaion, the baseline to be used is determined:

$B_c := \frac{1}{K_1}$ $B_c = 6.605 \cdot 10^3$ [m] Critical baseline, where decorrelation is unity

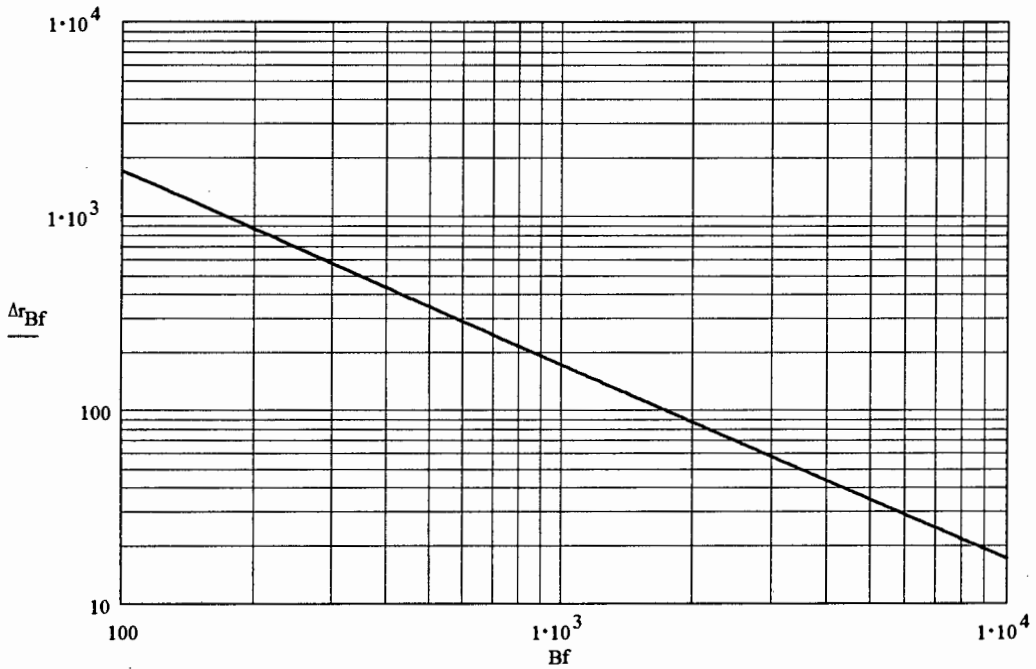
Range of acceptabe Baselines, corresponding to decorrelation of 0.2 to 0.6, and depending on the SNR:

$B_{shortest} := \frac{1 - (1 - 0.1) \cdot K_3}{K_1}$ $B_{shortest} = \begin{pmatrix} 1.201 \cdot 10^3 \\ 719.317 \\ 660.464 \end{pmatrix}$

Plot showing fringe spacing dependence on baseline length:

$Bf := 100, 110 .. 10000$ $\quad \alpha_1 := \frac{75.8}{360} \cdot 2 \cdot \pi$ $\quad \theta_b := \alpha_1$ $\quad \theta := 0$

$$\Delta r_{Bf} := \frac{\lambda \cdot r_1 \cdot \tan(\alpha_1 - \theta)}{Q \cdot Bf \cdot \cos(\alpha_1 - \theta_b)}$$



The optimum baseline from a height error point of view must now be traded off against the requirement for a fringe spacing that covers a certain number of range bins. At the look angle of 75.8 degrees, and optimum baseline length of 3000 m the spacing is 57m.

3. Terrain Shadowing and layover constraints:

A 90 degree range of slopes can be imaged by the radar. The look angle determines the location of the range of slopes imagable relative to a flat earth.

Steepest Slope imageable: $\theta_{max} := \alpha_1$ $\quad \theta_{max\_deg} := \frac{\theta_{max}}{2 \cdot \pi} \cdot 360$ $\quad \theta_{max\_deg} = 75.8$

Shallowest Slope imageable: $\theta_{min} := \alpha_1 - \frac{\pi}{2}$ $\quad \theta_{min\_deg} := \frac{\theta_{min}}{2 \cdot \pi} \cdot 360$ $\quad \theta_{min\_deg} = -14.2$

If a larger look angle is desired to increase the fringe spacing, then: $\alpha_1 := \frac{75}{360} \cdot 2 \cdot \pi$

Steepest Slope imageable: $\theta_{max} := \alpha_1$ $\quad \theta_{max\_deg} := \frac{\theta_{max}}{2 \cdot \pi} \cdot 360$ $\quad \theta_{max\_deg} = 75$

Shallowest Slope imageable: $\theta_{min} := \alpha_1 - \frac{\pi}{2}$ $\quad \theta_{min\_deg} := \frac{\theta_{min}}{2 \cdot \pi} \cdot 360$ $\quad \theta_{min\_deg} = -15$

WORKSHEET C

1. Parameter Sensitivity of the VHF SAR System

The parameters of the VHF SASAR system are:

$\lambda := 2.12$ $\quad \alpha_{10} := \frac{75.8}{360}\cdot 2\cdot\pi$ $\quad r_1 := 40766$ $\quad z_1 := 10000$

$f := 141\cdot 10^6$ $\quad c := 3\cdot 10^8$

The InSAR parameters chosen are:

$\theta_b := \alpha_{10}$ Look angle at centre of swath is set equal to baseline tilt

$\alpha_1 := \alpha_{10}, \frac{64}{63}\cdot\alpha_{10} .. \frac{75}{63}\cdot\alpha_{10}$

$Q := 2$

$B := 3953$ decorrelation of 0.4, and infinite SNR

The basic equations for the InSAR geometry are given as:

$$\phi(\alpha_1) := \frac{2\cdot\pi\cdot Q}{\lambda}\cdot B\cdot\sin(\alpha_1 - \theta_b)$$

$\alpha_1 := \alpha_{10}$

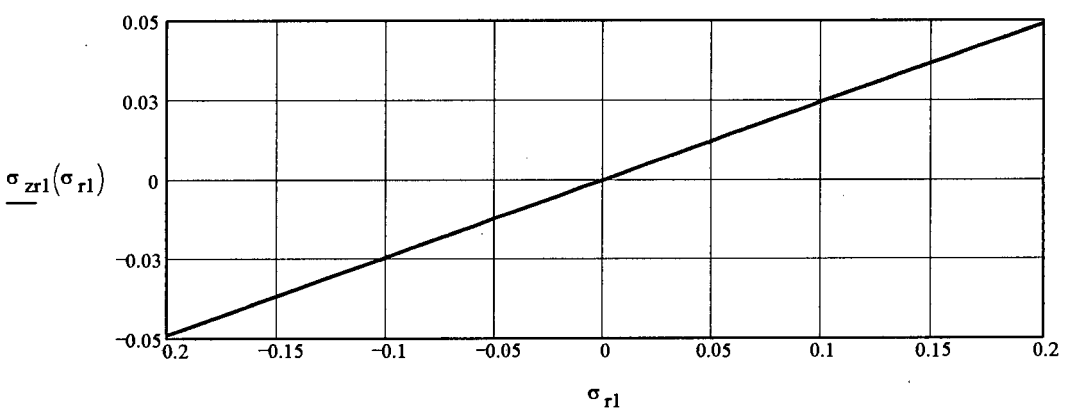
$z := z_1 - r_1\cdot\cos(\alpha_1)$

$y := r_1\cdot\sin(\alpha_1)$

The sensitivity equations are :

1. Sensitivity to measurement of slant range bin centre: $\sigma_{r1} := -0.2, -0.19 .. 0.2$

$$\sigma_{zr1}(\sigma_{r1}) := \cos(\alpha_1)\cdot\sigma_{r1}$$



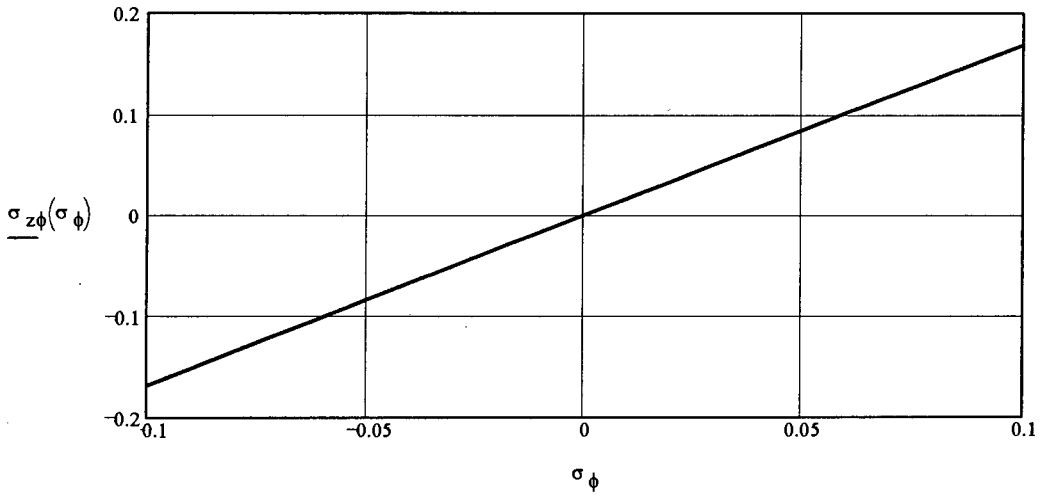
2. Sensitivity to aircraft height measurement: $\sigma_{z1} := -2, -1 .. 2$

$\sigma_{zz1}(\sigma_{z1}) := \sigma_{z1}$ An error in height measurement translates directly into a topographical error

3. Sensitivity to interferometric phase measurement: $\sigma_\phi := -0.1, -0.09 .. 0.1$

$$\sigma_{z\phi}(\sigma_\phi) := \frac{r_1 \cdot \sin(\alpha_1) \cdot \lambda}{2 \cdot \pi \cdot Q \cdot B \cdot \cos(\alpha_1 - \theta_b)} \cdot \sigma_\phi$$

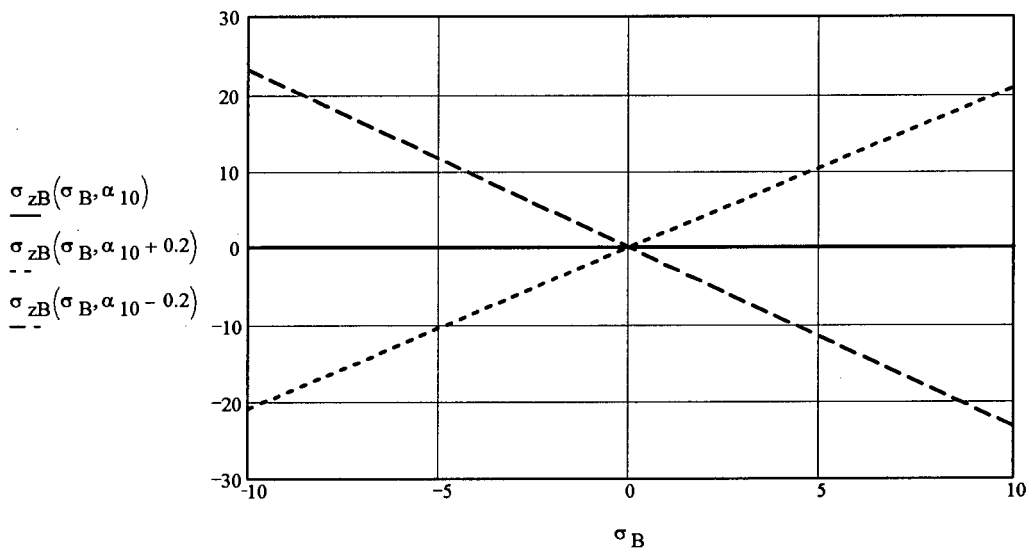
This is discussed in a separate section for the case at the centre of the swath. At that point, the error can be minimized by setting the baseline tilt angle equal to the look angle.



4. Sensitivity to measurement of the Baseline length: $\sigma_B := -10, -9 .. 10$

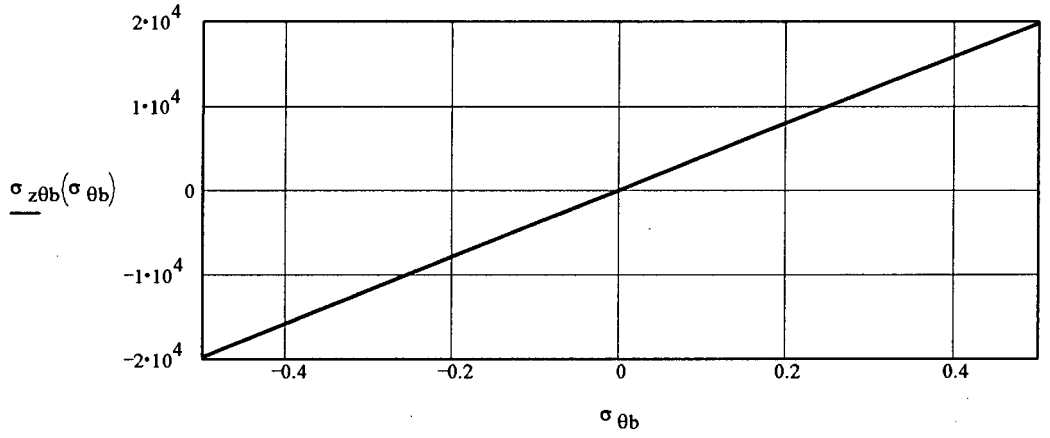
$$\sigma_{zB}(\sigma_B, \alpha_1) := \frac{r_1 \cdot \tan(\alpha_1 - \theta_b)}{B \cdot \sin(\alpha_1)} \cdot \sigma_B$$

At mid swath this error is set to zero by setting the baseline tilt angle equal to the look angle.



5. Sensitivity to baseline tilt angle: $\sigma_{\theta b} := -0.5, -0.4 .. 0.5$

$$\sigma_{z\theta b}(\sigma_{\theta b}) := r_1 \cdot \sin(\alpha_1) \cdot \sigma_{\theta b}$$

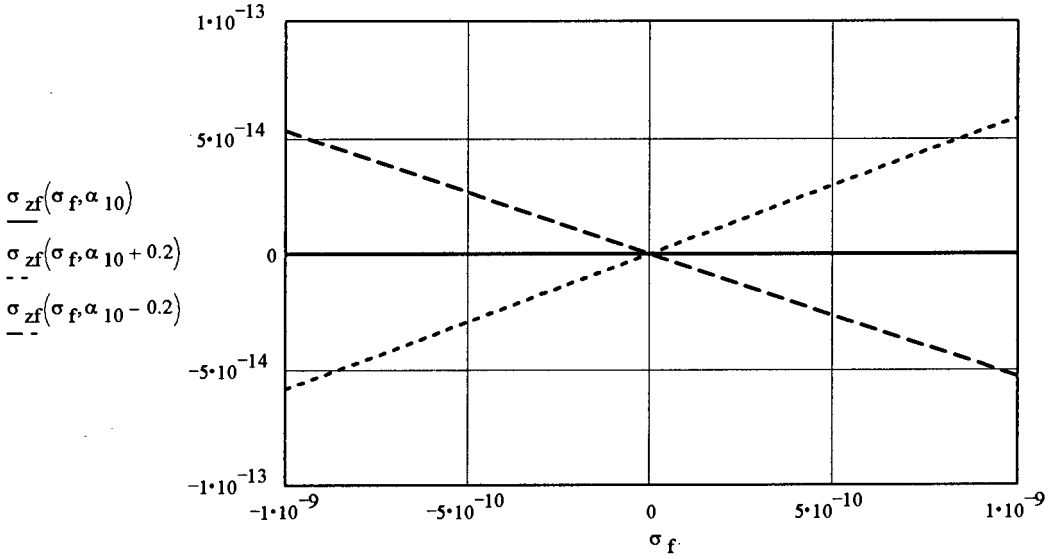


The InSAR system is extremely sensitive to angular inaccuracies. Uncompensated roll angles will directly influence accuracy through inaccurate tilt angle estimates.

6. Sensitivity to frequency (frequency stability): $\sigma_f := -10^{-9}, -0.9 \cdot 10^{-9} .. 10^{-9}$

$$\sigma_{zf}(\sigma_f, \alpha_1) := \frac{r_1 \cdot \phi(\alpha_1) \cdot \sin(\alpha_1) \cdot c}{2 \cdot \pi \cdot Q \cdot B \cdot \cos(\alpha_1 - \theta_b) \cdot f^2} \cdot \sigma_f$$

At mid swath this error is set to zero by setting the baseline tilt angle equal to the look angle.



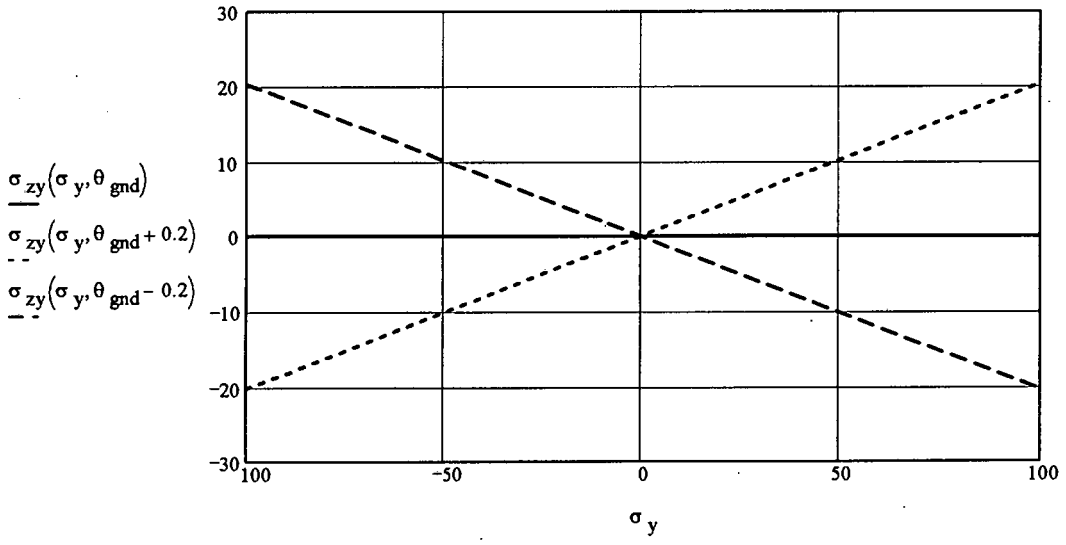
The InSAR system is extremely sensitive to frequncy errors, but these frequency errors can be kept extremely small.

7. Location induced errors due to misplacement of height: (zero for flat terrain) $\theta_{gnd} := 0$

$\sigma_y := -100, -90 .. 100$ $\sigma_x := -100, -90 .. 100$

$\sigma_{zy}(\sigma_y, \theta_{gnd}) := \tan(\theta_{gnd}) \cdot \sigma_y$ These errors are proportional to the slope of the terrain

$\sigma_{zx}(\sigma_x, \theta_{gnd}) := \tan(\theta_{gnd}) \cdot \sigma_x$



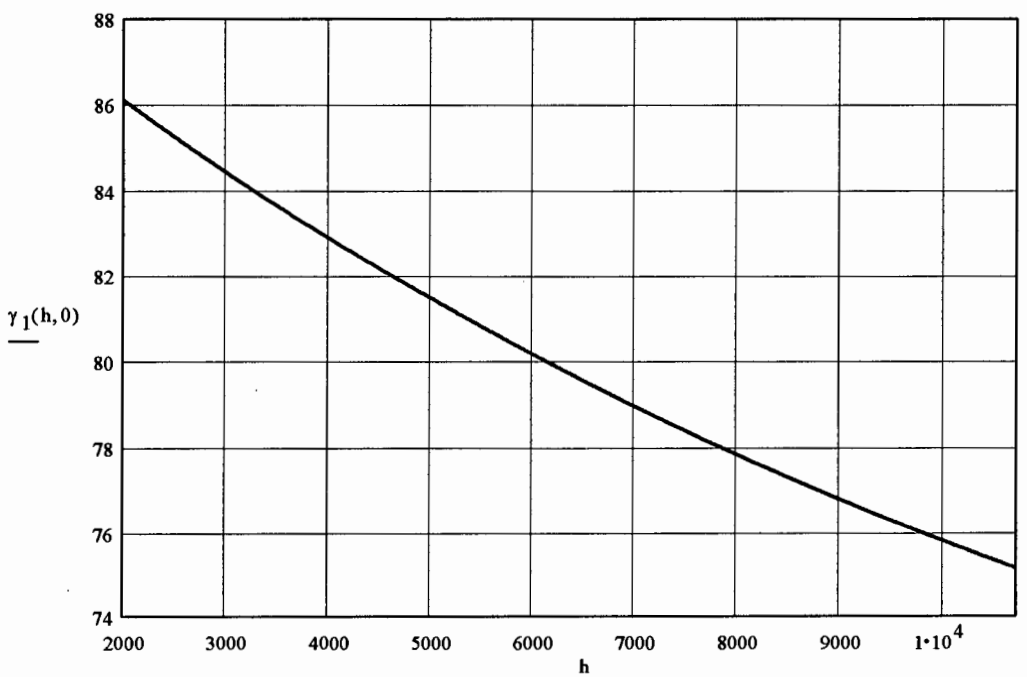
WORKSHEET D

1. SASAR incidence angle Dependency on Altitude at Mid Swath

$h := 2000, 2010 .. 10700$	Altitude range of SASAR [m]
$Pixels := 4096$	Number of pixels in swath
$\delta R_s := 26$	Slant range resolution [m]
$n_R := 13$	Slant range sampling [m]
$\theta_t := 0$	Terrain Slope
$\lambda := 2.13$	Wavelength [m]

$$\gamma_1(h, \theta_t) := \operatorname{acos}\left(\frac{1}{\sqrt{2 + \frac{Pixels \cdot n_R}{2 \cdot h}}}\right) \cdot \frac{360}{2 \cdot \pi} - \theta_t$$

Incidence angle at mid swath as a function of altitude



For a given terrain slope, the incidence angle is the value in thegraph, minus the terrain slope angle.

2. Baseline Length determined by incidence angle and required samples per fringe

$n := 2$ Required samples per fringe

$$r_1(\gamma_f) := \frac{1}{1 - \sqrt{2}\cdot\cos(\gamma_f)}\cdot\frac{\text{Pixels}}{2}\cdot n\,R$$

Slant range at mid swath for the range of incidence angles for flat terrain of 75.8 degrees at 10000 m to 86.1 degrees at 2000 m altitude

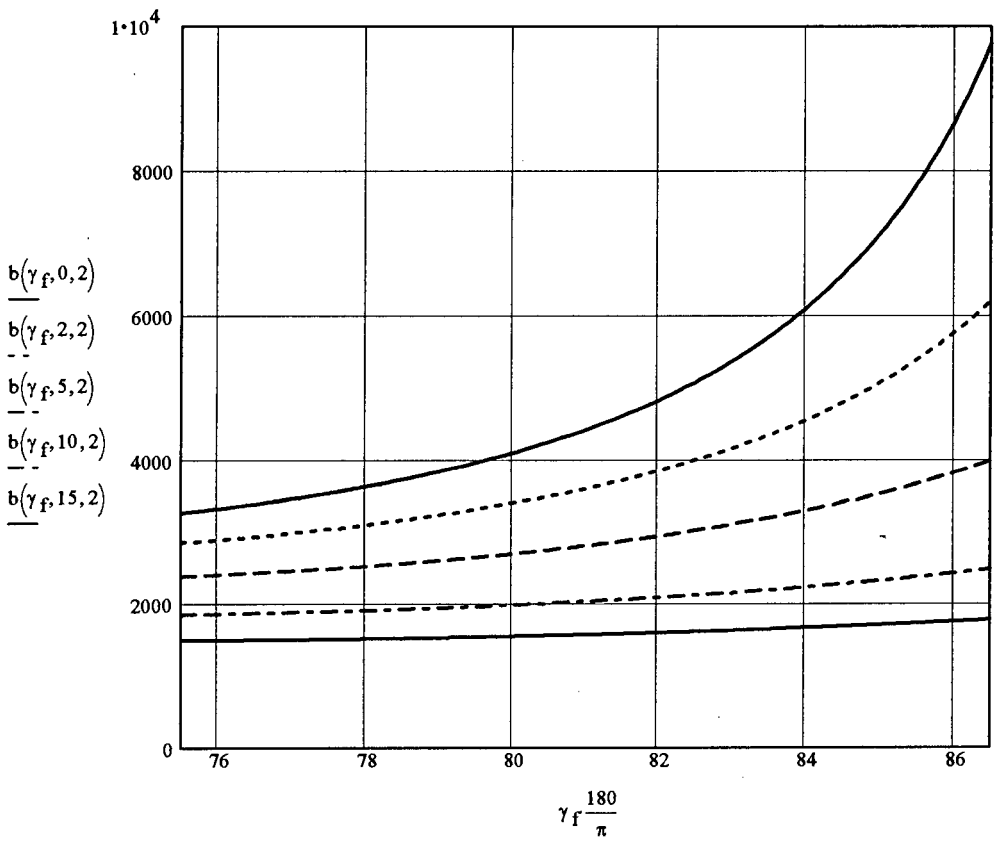
$$\gamma_f := 75.5\cdot\frac{\pi}{180}, 75.6\cdot\frac{\pi}{180} .. 86.5\cdot\frac{\pi}{180}$$

Incidence angle for flat terrain

$$\gamma_2(\gamma_f, \theta_t) := \gamma_f - \theta_t\cdot\frac{\pi}{180}$$

$\theta_1(\gamma_f) := 0$ at mid swath if the baseline is perpendicular to the look direction

$$b(\gamma_f, \theta_t, n) := \frac{\lambda\cdot r_1(\gamma_f)\cdot\tan(\gamma_2(\gamma_f, \theta_t))}{2\cdot n\cdot\delta R_s\cdot\cos(\theta_1(\gamma_f))}$$



3. Acceptable Baseline lengths in terms of Decorrelation

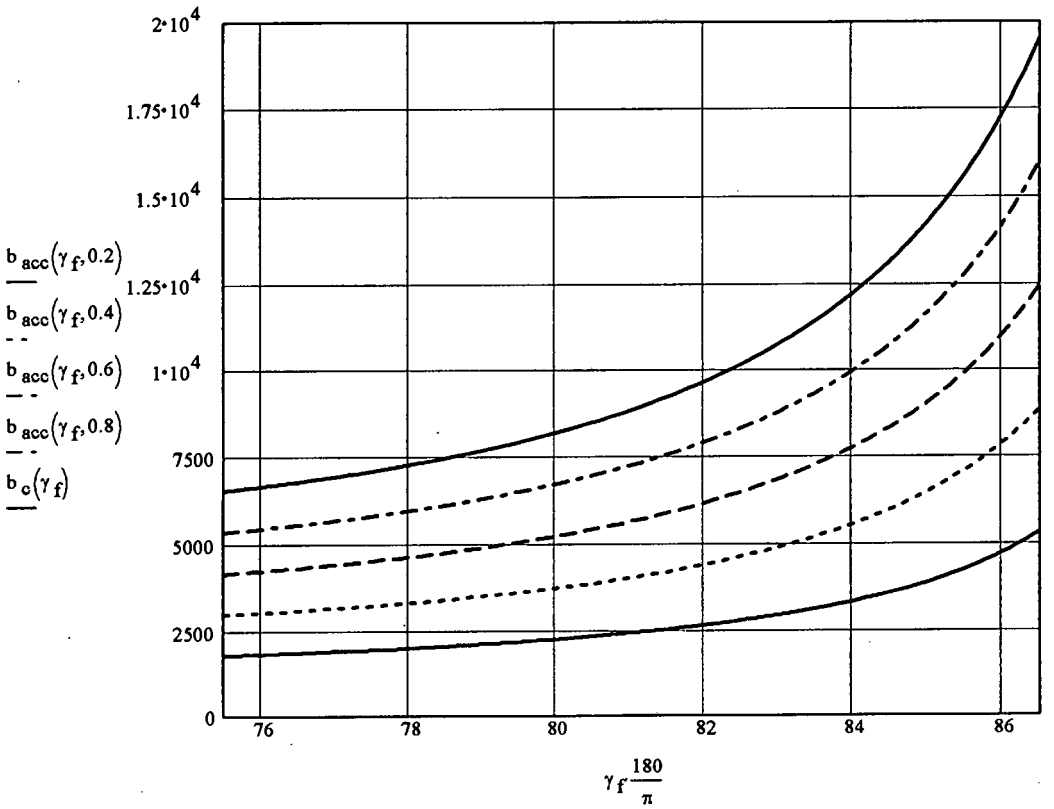
$Q := 2$ Repeat-pass system

$dc := 0.2, 0.3 .. 0.8$ decorrelation

$SNR := 10$ 10 dB SNR

$$b_c(\gamma_f) := \frac{\lambda \cdot \tan(\gamma_f)}{Q \cdot \delta R_s} \cdot \frac{Pixels \cdot n_R}{2} \cdot \left(\frac{1}{1 - \sqrt{2} \cdot \cos(\gamma_f)} \right)$$

$$b_{acc}(\gamma_f, dc) := b_c(\gamma_f) \cdot \left(1 - \frac{1 - dc}{1 + \frac{1}{SNR}} \right)$$



4. Maximum Imagable Slopes over the Operating Envelope

$$h(\gamma_f) := \frac{Pixels \cdot n_R}{2 \cdot \left(\frac{1}{\cos\left(\gamma_f \frac{\pi}{180}\right)} - \sqrt{2} \right)} \qquad h(75.8) = 1 \cdot 10^4$$

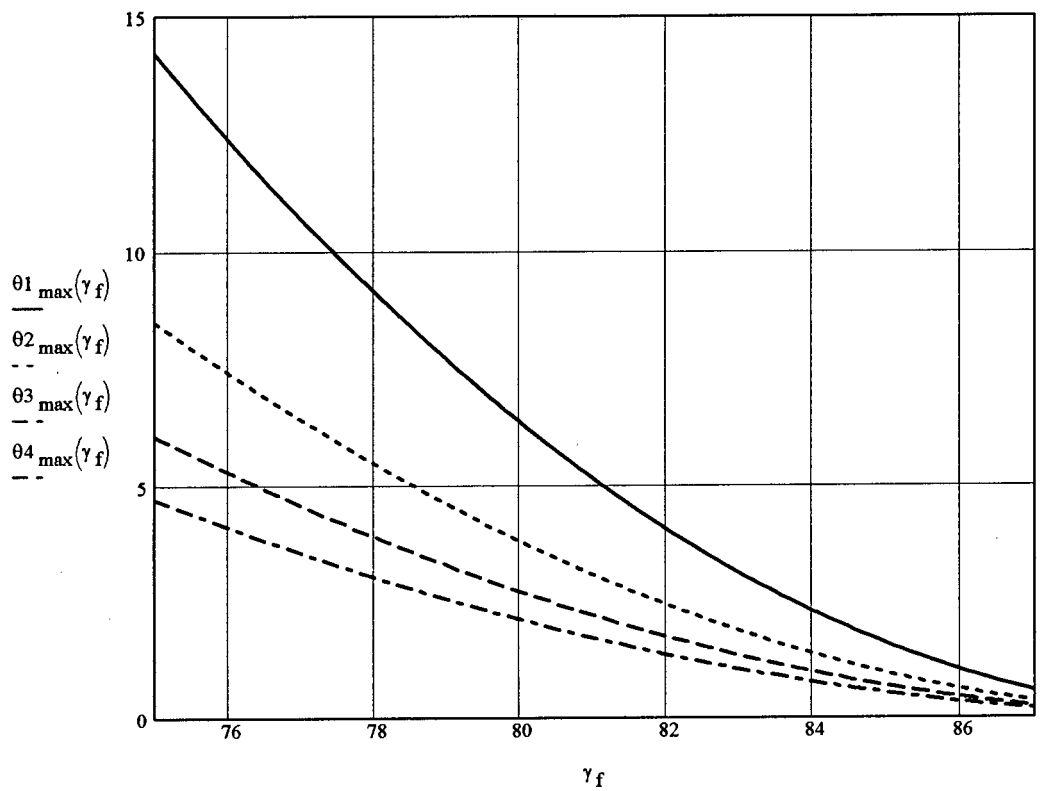
$$\theta 1_{max}(\gamma_f) := \operatorname{asin}\left(\frac{\lambda}{2 \cdot Q \cdot n_R} \cdot \frac{h(\gamma_f)}{b_{acc}\left(\gamma_f \frac{\pi}{180}, 0.2\right)} \cdot \sin\left(\gamma_f \frac{\pi}{180}\right) \right) \cdot \left(\frac{180}{\pi}\right) \qquad b_{acc}\left(75.8 \cdot \frac{\pi}{180}, 0.2\right) = 1.8 \cdot 10^3$$

$\theta 1_{max}(75.8) = 12.747$ $\qquad \gamma_f := 75, 75.1 .. 87$

$$\theta 2_{max}(\gamma_f) := \operatorname{asin}\left(\frac{\lambda}{2 \cdot Q \cdot n_R} \cdot \frac{h(\gamma_f)}{b_{acc}\left(\gamma_f \frac{\pi}{180}, 0.4\right)} \cdot \sin\left(\gamma_f \frac{\pi}{180}\right) \right) \cdot \left(\frac{180}{\pi}\right)$$

$$\theta 3_{max}(\gamma_f) := \operatorname{asin}\left(\frac{\lambda}{2 \cdot Q \cdot n_R} \cdot \frac{h(\gamma_f)}{b_{acc}\left(\gamma_f \frac{\pi}{180}, 0.6\right)} \cdot \sin\left(\gamma_f \frac{\pi}{180}\right) \right) \cdot \left(\frac{180}{\pi}\right)$$

$$\theta 4_{max}(\gamma_f) := \operatorname{asin}\left(\frac{\lambda}{2 \cdot Q \cdot n_R} \cdot \frac{h(\gamma_f)}{b_{acc}\left(\gamma_f \frac{\pi}{180}, 0.8\right)} \cdot \sin\left(\gamma_f \frac{\pi}{180}\right) \right) \cdot \left(\frac{180}{\pi}\right)$$



Bibliography

[1] Richard Bamler and Dieter Just. Phase Statistics and Decorrelation in SAR Interferograms . *IGARSS '93, Tokyo, Japan*, 3:980 – 984, August 1993.

[2] D Barker. Characterisation of Short Range SAR - Phase 2 . Technical report, Defence Research Agency, Farnborough, Hampshire, GU146TD, U Kingdom, June 1993.

[3] D.K. Barton. *Radar Evaluation Handbook* . Artech House, Norwood, MA, 1990.

[4] D. Blacknell and Shaun Quegan. SAR motion compensation using autofocus . *Int. J. Remote Sensing*, 12(2):253 – 275, 1991.

[5] N. Brousseau. Focusing properties of synthetic aperture radar interferograms having range curvature aberrations. *Applied Optics*, 18(15):2580–2585, August 1979.

[6] Richard E. Carande. Dual Baseline and Frequency Along-Track Interferometry . *IGARSS '92*, pages 1585 – 1588, May 1992.

[7] Richard E. Carande. Estimating Ocean Coherence Time Using Dual-Baseline Interferometric Synthetic Aperture Radar. *IEEE Trans on Geosc. and Remote Sensing*, 32(4):846–854, July 1994.

[8] R A Cordey. Terrain Height Mapping Using Interferometric Satellite Radar Data . *IEEE*, 1993.

[9] Robert Crida. Reproduction of Shuttle Radar Images (SRI Software V3). Technical report, UCT Radar Remote Sensing Group, November 1991.

[10] Robert Crida. Reproduction of Shuttle Radar Images . Technical Report rrsg:10:93, UCT Radar Remote Sensing Group, June 1993.

[11] I G Cumming, David Hawkins, and A. Laurence Gray. All-Weather Mapping with Interferometric Radar . *Proc. 23rd Int. Symposium on Remote Sensing of Environment*, pages 1249–1262, April 1990.

[12] Marco de Fazio and Francesco Vinelli. DEM Reconstruction in SAR Interferometry: Practical Experiences with ERS-1 SAR Data. *IGARSS '93, Tokyo, Japan*, 3:1207 – 1209, August 1993.

[13] Gitta Domik. Methods and Application of Surface Shape Reconstruction From Multiple SAR Images. *IGARSS '87*, pages 1551–1555, May 1987.

[14] K.E. Dowson. Technical Specification for the Processing and Recording System of the SASAR. Technical report, Reutech Radar Systems, Technopark Stellenbosch, March 1995.

[15] Diane L. Evans, Tom G. Farr, Howard A. Zebker, Jakob J. van Zyl, and Peter J. Mouginis-Mark. Radar Interferometry Studies of the Earth's Topography . *EOS, Transactions, American Geophysical Union*, 73(52):553–560, December 1992.

[16] J. Patrick Fitch. The Single Antenna Interferometer . *IGARSS '91*, 4:2573 – 2576, 1991.

[17] G Franceschetti, M. Migliaccio, D. Riccio, and G Schirinzi. SARAS: A Synthetic Aperture Radar (SAR) Raw Signal Simulator . *IEEE Trans on Geosc. and Remote Sensing*, 30(1):110–123, January 1992.

[18] Anthony Freeman and Manfred Zink. An Application of the Monopulse Principle to Determining Elevation Angles in SAR Images. *IEEE Trans on Geosc. and Remote Sensing*, 32(3):616–625, May 1994.

[19] Andrew K. Gabriel and Richard M. Goldstein. Crossed orbit interferometry: theory and experimental results from SIR-B. *Int. J. Remote Sensing*, 9(5):857 – 872, 1988.

[20] Andrew K. Gabriel, Richard M. Goldstein, and Howard A. Zebker. Mapping Small Elevation Changes Over Large Areas: Differential Radar Interferometry. *Journal of Geophysical Research*, 94(B7):9183–9191, July 1989.

[21] Fabio Gatelli, Andrea Monti Guarnieri, Francesco Parizi, Paoli Pasquali, Claudio Prati, and F. Rocca. The Wavenumber Shift in SAR Interferometry . *IEEE Trans on Geosc. and Remote Sensing*, 32(4):855–865, July 1994.

[22] D.C. Ghiglia and Gary A. Mastin. Two-dimensional phase correction of synthetic-aperture-radar imagery. *Optics Letters*, 14(20):1104–1106, October 1989.

[23] D.C. Ghiglia and 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*, 11(1):107–117, January 1994.

[24] P. Golda. *Manual for Radar Simulator Program as at June 1995* , June 1995.

[25] Richard M. Goldstein and Howard A. Zebker. Interferometric radar measurement of ocean surface currents . *Nature*, 328:707–709, August 1987.

[26] Richard M. Goldstein, Howard A. Zebker, and Charles L. Werner. Satellite radar interferometry: Two-dimensional phase unwrapping . *Radio Science*, 23(4):713–720, 1988.

[27] Leroy C. Graham. Synthetic Interferometer Radar for Topographic Mapping . *Proceedings of the IEEE*, 62(6):763–768, June 1974.

[28] A. Laurence Gray and Peter J. Farris-Manning. Repeat-Pass Interferometry with Airborne Synthetic Aperture Radar . *IEEE Trans on Geosc. and Remote Sensing*, 31(1):180–191, January 1993.

[29] Hugh D Griffiths and Andrew J Wilkinson. Improvements in phase unwrapping algorithms for interferometric SAR. *L'Onde Electrique*, 74(3):46–52, June 1994.

[30] J.M. Horrell and M.R. Inggs. UCT SAR C Processor Kernel - User's Manual . Technical report, UCT Radar Remote Sensing Group, March 1995.

[31] J.M. Horrell and M.R. Inggs. UCT SAR C Processor Kernel Report . Technical report, UCT Radar Remote Sensing Group, March 1995.

[32] S.A. Hovanessian. *Radar System Design and Analysis* . Artech House, Norwood, MA, January 1984.

[33] M.R. Inggs. SASAR Data Handling System . Technical Report 1158:95, UCT Radar Remote Sensing Group, March 1995.

[34] M.R. Inggs. Status of the SASAR System . In *Proceedings of the IEEE 1996 Geoscience and Remote Sensing Symposium*, Lincoln, University of Nebraska, Lincoln, May 1996. IEEE 1996 Geoscience and Remote Sensing Symposium.

[35] M.R. Inggs. The SASAR VHF Sensor . In *Proceedings of European Conference on Synthetic Aperture Radar 1996*, page 4, Konigswinter, Germany, March 1996. EUSAR '96.

[36] M.R. Inggs and J.M. Horrell. *The SASAR Airborne SAR System Specification, Ver:1.1* , March 1995.

[37] Naoya Kobayashi. Terrain height measurement by synthetic aperture radar with an interferometer. *Int. J. Remote Sensing*, 7(3):339–348, 1985.

[38] Jong-Sen Lee, K.W. Hoppel, and A.R. Miller. Intensity and Phase Statistics of Multilook Polarimetric and Interferometric SAR Imagery. *IEEE Trans on Geosc. and Remote Sensing*, 32(5):1017–1028, September 1994.

[39] Fuk K. Li and Richard M. Goldstein. Studies of Multibaseline Spaceborne Interferometric Synthetic Aperture Radars. *IEEE Trans on Geosc. and Remote Sensing*, 28(1):88–97, January 1990.

[40] Qian Lin and John F Vesecky. Topography Estimation with Interferometric Synthetic Aperture Radar using Fringe Detection. *IGARSS '91*, pages 2173 – 2176, June 1991.

[41] Qian Lin, John F Vesecky, and Howard A. Zebker. New Approaches in Interferometric SAR Data Processing . *IEEE Trans on Geosc. and Remote Sensing*, 30(3):560–567, May 1992.

[42] Qian Lin, John F Vesecky, and Howard A. Zebker. Phase unwrapping through least squares approximation . *Proceedings of the SPIE*, 1657:244–255, February 1992.

[43] Qian Lin, John F Vesecky, and Howard A. Zebker. Phase unwrapping through fringe-line detection in synthetic aperture radar interferometry. *Applied Optics*, 33(2):201–208, January 1994.

[44] Soren N. Madsen and Howard A. Zebker. Automated Absolute Phase Retrieval in Across-track Interferometry . *IGARSS '92*, 2:1582–1584, 1992.

[45] Soren N. Madsen, Howard A. Zebker, and Jan Martin. Topographic Mapping Using Radar Interferometry: Processing Techniques. *IEEE Trans on Geosc. and Remote Sensing*, 31(1):246–256, January 1993.

[46] M Marom, Richard M. Goldstein, E.B. Thornton, and L. Shemer. Remote Sensing of ocean wave spectra by interferometric synthetic aperture radar. *Nature*, 345:793–795, June 1990.

[47] Didier Massonnet and Thierry Rabaute. Radar Interferometry: Limits and Potential . *IEEE Trans on Geosc. and Remote Sensing*, 31(2):455–464, March 1993.

[48] A.V. Oppenheim and Jae S. Lim. The Importance of Phase in Signals . *IEEE Proceedings*, 69(5):529–541, May 1981.

[49] Francesco Parizi, Andrea Monti Guarnieri, C. Prati, and F. Rocca. SAR Interferometry Experiments with ERS-1 . *IGARSS '94*, pages 991–993.

[50] C. Prati and F. Rocca. Limits to the resolution of elevation maps from stereo SAR images . *International Journal of Remote Sensing*, 11(12):2215–2235, 1990.

[51] Claudio Prati. 3-D Synthetic Aperture Radar Surveys . *Internat Conference on Acoustics, Speech and Signal Processi*, 4:2376–2379, May 1989.

[52] Claudio Prati, Fabio Rocca, and Andrea Monti Guarnieri. Effects of Speckle and Additive Noise on the Altimetric Resolution of Interferometric SAR (ISAR) Surveys. *IGARSS '89*, pages 2469 – 2472, 1989.

[53] R. Keith Raney and Paris W. Vachon. Synthetic Aperture Radar Imaging of Ocean Waves From an Airborne Platform: Focus and Tracking Issues. *Journal of Geophysical Research*, 93(C10):12475–12486, October 1988.

[54] R. Keith Raney and Paris W. Vachon. A Phase Preserving SAR Processor . *IGARSS '89*, pages 2588 – 2591, 1989.

[55] E Rodriguez and J.M. Martin. Theory and design of interferometric synthetic aperture radars . *IEEE Proceedings*, 139(2):147–159, April 1992.

[56] M. S. Scivier and M.A. Fiddy. Phase ambiguities and the zeros of multidimensional band-limited functions. *Optical Society of America*, 2(5):693–697, May 1985.

[57] L. Shemer, M Marom, and D Markman. Estimates of Currents in the Nearshore Ocean Region using Interferometric Synthetic Aperture Radar. *Journal of Geophysical Research*, 98(C4):7001–7010, April 1993.

[58] Merril I Skolnik. *Introduction to Radar Systems, 2nd Edition* . McGraw-Hill Book Company, New York, January 1980.

[59] David Small, Charles L. Werner, and Daniel Nuesch. Baseline Modelling for ERS-1 SAR Interferometry . *IGARSS '93, Tokyo, Japan*, 3:1204 – 1206, August 1993.

[60] D.R. Stevens, I G Cumming, and M R Ito. Airborne interferometric SAR: Terrain Induced Phase Errors . *IGARSS '93, Tokyo, Japan*, 3:977 – 979, August 1993.

[61] George W Stimson. *Introduction to Airborne Radar* . Hughes Aircraft Company, El Segundo, California, USA, 1983.

[62] F.G. Stremler. *Introduction to Communication Systems: 3rd Edition* . Addison-Wesley Publishing Company, Reading, MA, USA, 1990.

[63] Hiroaki Takajo and Tohru Takahashi. Least-squares phase estimation from the phase difference . *Journal of the Optical Society of America, A.*, A5(3):416–425, March 1988.

[64] Hiroaki Takajo and Tohru Takahashi. Noniterative method for obtaining the exact solution for the normal equation in least-squares phase estimation from the phase difference. *Journal of the Optical Society of America, A.*, 5(11):1818–1827, November 1988.

[65] Donald R Wehner. *High Resolution Radar* . Artech House, Norwood, MA, January 1987.

[66] Andrew J Wilkinson. Development of an Interferometric SAR Processing Facility Phase 1: Theoretical Principles and Algorithms. Technical report, UCT Radar Remote Sensing Group, March 1996.

[67] Howard A. Zebker. Decorrelation in Interferometric Radar Echoes . *IEEE Trans on Geosc. and Remote Sensing*, 30(5):950–959, September 1992.

[68] Howard A. Zebker and Richard M. Goldstein. Topographic Mapping From Interferometric Synthetic Aperture Radar Observations. *Journal of Geophysical Research*, 91(B5):4993–4999, April 1986.

[69] Howard A. Zebker, Soren N. Madsen, Jan Martin, Kevin B. Wheeler, Timothy Miller, Yunling Lou, Giovanni Alberti, Sergio Vetrella, and Alessandro Cucci. The TOPSAR Interferometric Radar Topographic Mapping Instrument . *IEEE Trans on Geosc. and Remote Sensing*, 30(5):933–940, September 1992.

[70] Howard A. Zebker, Charles L. Werner, Paul A. Rosen, and Scott Hensley. Accuracy of Topographic Maps Derived from ERS-1 Interferometric Radar. *IEEE Trans on Geosc. and Remote Sensing*, 32(4):823–836, July 1994.