

Phase Unwrapping Algorithm Using Edge Detection and Statistical Cost Functions

MSC GEOMATICS
THESIS



NARESH SONI
AUGUST 2012

DEPARTMENT OF GEOSCIENCE & REMOTE SENSING
FACULTY OF CIVIL ENGINEERING AND GEOSCIENCES

Supervisor	Dr. Andrew Hooper
Graduation professor	Prof.dr.ir R.F. Hanssen
Co-supervisor	Sami Samiei Esfahany(MSc.)
Co-reader	dr. ir. H. van der Marel
External Examiner	ir. Edward Verbree

Abstract

Two-dimensional (2-D) phase field is a significant output of several remote sensing applications including interferometric synthetic aperture radar (InSAR). This field is a modulo 2π phase value of observed signal which means that the values are offset in an interval resulting in measurements as a fraction of a complete cycle. This phase field is termed as wrapped phase. Phase unwrapping is the process of removing this ambiguity and reconstructing the phase field for better interpretation of the data. Phase unwrapping is one of the most prominent research topics in digital image processing and extensive work has been done in the field during the last couple of decades. 2-D phase unwrapping, in particular, is a key step and among the most significant sources of error in the analyses of SAR interferograms. Many novel techniques have been developed for 2-D phase unwrapping, most of which are formulated as a mathematical optimization problem. These techniques serve their purpose in cases where the signal is coherent and smooth, but perform poorly in noisy and faulty regions. SNAPHU, in particular, is a very elegant algorithm which relies on statistical cost-functions and network-flow optimization techniques. SNAPHU provides a higher degree of flexibility in setting up the unwrapping problem since the cost-functions are generalized objective functions which can be adjusted according to requirements. SNAPHU performs well in general but sometimes provides poor results in the regions with discontinuities. That is why efforts have been made during the current research work to generate a new algorithm, which relies on the SNAPHU solver routine and at the same time improves results by enhancing cost-functions with additional parameters. In the best case scenario, the new algorithm should improve results. On the other hand, if the new parameters are not reliable, it should produce results which are as good

as the existing algorithm.

The purpose of the current study is to assess various phase unwrapping algorithms (particularly SNAPHU) and then develop a method to improve the cost function. The study is divided mainly into two parts. The first part deals with the extraction of additional parameters, which reflects the phase gradient in local neighborhood. The aim is to locate edges where phase gradient is changing abruptly at the same time estimate phase gradients in smoother regions of the interferogram. This step involves the use of signal processing techniques (Windowed Fourier Transformation). Standard edge detection techniques are used to finalize the output from this stage. It is observed that edges can be detected in wrapped phase field and accuracy improves if the input parameters are set up on a case basis. The second step is to efficiently include these edges in terms of cost-functions such that unwrapped results improve in both faulty and non-faulty regions.

The new proposed algorithm has a good theoretical basis but the goal is to implement a time efficient solver routine. The algorithm has been tested on simulated data and has performed well. Real life interferograms are not thoroughly investigated and only a few examples have been shown to prove the workability of the algorithm. The computation time is reasonable in cases where the size of the data is small; however under given time constraints of thesis duration, the algorithm remains computationally expensive for large data sets. The computational time greatly improves if prior information is used to set up the problem.

Preface

This thesis work is done for the purpose of completion of Master of Science (Geomatics) degree at TU Delft, The Netherlands. By the end of my first year, I was looking for a good master's thesis topic. During *Microwave remote sensing* class, Prof. Ramon Hanssen expressed his view that his section has some good master's thesis options. I pursued the idea of doing my research work at Mathematical Geodesy and Positioning section which later became the Department of Geoscience and Remote sensing at the Faculty of Civil Engineering and Geosciences. I was particularly interested in work where I would be more involved with theoretical and mathematical analysis. After some talks, Dr. Andrew Hooper, supervisor of my thesis work, proposed a topic related to phase unwrapping algorithms. I studied this topic during my course work and this field appealed to me. Finally, I started working on it by early October 2011.

With respect to Geomatics masters, Radar remote sensing covers the complete spectrum extensively i.e. data acquisition, processing and application of Geo-information. The capabilities of the field have a wide gamut and new possibilities are coming up. So far, it has been used for geodetic measurement of the natural environment particularly deformation monitoring of surface over large areal extents caused by several geological activities such as earthquakes, volcanoes etc. But more recent research indicates that it can be a useful technique for monitoring of the urban environment too. The future of the field is promising and that is one of the motivations behind the current research work. The thesis is more focused on the data processing part of geo-information. Data is acquired by a satellite and after some initial processing phase unwrapping comes into the picture, which is the focus of this thesis.

Thesis work within MSc(Geomatics) carries a work load of 45 ECTS and is equivalent to 1260 hours of study. The study lasted for approximately 9-10 months.

Acknowledgements

I would like to take this opportunity to thank all the people who motivated and guided me during my masters. First of all, I would to thank Dr. Andrew Hooper for his great supervision. His critical but supportive approach guided the work in the desired direction. He continuously challenged me to make further efforts to improve the quality of work. Personally, I learnt a lot in terms of time management and developing a keen eye towards the subtleties of the problem. I would like to thank Prof. dr. ir. Ramon Hanssen, for developing my interest in the field and explaining complex remote sensing fundamentals in the easiest possible way. I am deeply indebted to my daily supervisor, Sami Samiei-Esfahany, who helped me with all the important steps of the research work, from development to implementation and writing. 'Thinking-out-loud' sessions helped in exploring several possibilities towards finding a solution. I am greatly thankful to my friends at Delft. Special thanks to Joana, Pooja and Gargi for good company during small panic attacks I had during thesis work. Very special thanks to Prof. Sanjay Pareek, Dr. Anupam, Mrs. Minakshi and Amya for providing me with a homely environment. They made a home away from home in Delft.

Finally, and most importantly, I would to thank my parents and my sisters, for their continuous love, support and encouragement throughout my stay in Delft.

Naresh Soni

Delft, The Netherlands

August, 2012

Contents

Abstract	iii
Preface	v
1 Introduction	3
1.1 Background	4
1.2 Problem formulation and research objective	6
1.3 Scope of study	7
1.4 Outline	7
2 Phase Unwrapping Algorithms	9
2.1 Introduction	9
2.2 General method and assumption	10
2.2.1 Nyquist criterion	11
2.2.2 Reconstruction of phase from phase gradients	11
2.2.3 Conservative and non-conservative field	13
2.3 Previous formulations and Algorithms	15
2.3.1 Branch-cut algorithm	16
2.3.2 L^p norm optimization	19
2.3.3 Network-flow algorithms	21
3 Statistical Models and Non-linear Optimization	25
3.1 MAP distribution functions	26
3.2 Deformation and MAP cost functions	29
3.3 SNAPHU solver routine	31

3.4	Analysis of unwrapping algorithm using SNAPHU	32
4	Edge detection and Cost function generation	43
4.1	Windowed Fourier Transformation	44
4.1.1	Windowed Fourier filtering(WFF)	45
4.1.2	Windowed Fourier Ridges	49
4.2	Edge detection and processing	51
4.3	New cost functions	52
5	Results and discussions	55
5.1	Testing with simulated data	55
5.2	Testing with real data	58
5.3	Limitations	62
6	Conclusion and Recommendations	65
6.1	Conclusion	65
6.2	Recommendations	66
A	SNAPHU configuration file	69

List of Figures

1.1	An example of an interferogram of Izmit earthquake occurred in 1999. The fault occurred during this earthquake is shown as a line.(Credits: NASA/University of Oxford)	4
2.1	Example of a simulated wrapped and unwrapped phase field. The scale is in radians. Wrapped phase field is obtained by removing inter number of cycles from unwrapped phase.	10
2.2	Coordinate convention followed in the current thesis. (i, j) are in accordance with standard matrix coordinate system. (x, y) are image coordinates.	12
2.3	Example of 1-D wrapped and unwrapped array. (a) Wrapped phase values shown as a fraction of a cycle (b) Unwrapped phase values obtained under Nyquist criterion.	13
2.4	Irrotational field. The δ values are phase gradients, in clock-wise order. The summation of these gradients is equal to zero in an irrotational field.	14
2.5	Path dependence in irrotational field. (a) The 2x2 pixel phase field is not conservative since the sum of residues does not vanish. (b) and (c) are unwrapped solutions, but integration is done by different paths. As observed, the final unwrapped result is not the same and depends on choice of path.	15

2.6	Example of a branch-cut. The phase values are shown as a fraction of a cycle. Residues are mostly 0. Positive and negative residues are connected by a line. This line serves as barrier and integration is not allowed over this cut.	17
2.7	Nodes and arcs representation of a phase field. a,b,c,d are pixels. E is an arc and V is node. A complete phase field can be represented in terms of nodes and arcs accordingly.	18
2.8	Example of an MST and MCF solution. Blue line is an MST solution where the total cost of the flow is 12. Red line is one of the probable solutions given by MCF optimization where total cost of flow is 6.	23
3.1	Example of an unwrapped gradient PDF. $f(\Delta\phi)$ takes only values which are integer offset from wrapped gradient $\Delta\phi^w$ under the assumption of congruence. The shape of PDF is general and doesn't represent the real shape of MAP PDFs used in SNAPHU (Chen, 2001).	27
3.2	The MAP cost functions in both cases where $\gamma \geq \gamma_{min}$ and $\gamma \leq \gamma_{min}$. Solid line is for the case where $\gamma < \gamma_{min}$. Otherwise its a dashed line (Chen, 2001).	31
3.3	Example of simulated data. (i)Top left is unwrapped phase (ii) Top right is wrapped phase (iii) Bottom left is same unwrapped phase but 30° standard deviation noise is also added (iv) similarly wrapped phase with noise. Scale is in radians.	33
3.4	Phase jumps in the simulated data. These are the locations where phase gradient is more than half a cycle. These locations are obtained from true unwrapped solution and are shown on a wrapped solution.	34
3.5	Residues in the simulated data are shown as black and white dots. Black dots correspond to -1 and white dots correspond to $+1$ residue. These are the locations where closed loop integral of phase gradients is not vanishing. It is observed that residues generally occur in pairs.	35

3.6	Branch cuts made by SNAPHU. Dark red line is the location of the real fault where there are true phase jumps. Black lines are branch cuts estimated by SNAPHU. It can be observed that SNAPHU branch cuts don't represent the real location of the discontinuity.	36
3.7	Error of the unwrapped solution generated by SNAPHU with respect to true unwrapped solution. Scale is number of cycles.	37
3.8	Number of pixels with error values. x -axis is number of cycles.	37
3.9	Example of cost function parameters. The figure is not to scale and provides a general idea about the shape of the objective functions. Number 1-4 represent the columns as shown in Table: 3.1	38
3.10	The unwrapped solution generated by SNAPHU using external cost functions which are generated using true simulated unwrapped solution. Since the locations of true phase jumps are known, function is set such that it is more probable to have a branch-cut at the location of true phase jumps. Using this information, cost functions are generated which provides improved results.	40
3.11	The error of the unwrapped solution using externally created cost functions.	40
3.12	The number of pixels with error values of the unwrapped solution using manually created cost functions.	41
4.1	Example of simulated wrapped phase. Size is 400×400 . Area has a fault running across it. Rest of the interferogram is smooth. 20° standard deviation noise is added.	47
4.2	The Fourier Correlation Coefficient map of Fig. 4.1. Scale is (0,1). The regions with the fault have smaller values and regions with a pattern have higher.	47
4.3	Edges detected by Canny Edge detector in Fig. 4.2. The amplitude is low in a band near the fault so 2 lines are detected. Post processing is required to finally make one edges out of 2 lines. The corners of the interferograms are also detecting edges because of insufficient sampling. These edges need to be dropped.	48

4.4	Local frequencies in X direction using WFR technique. Scale is radians per pixel in x direction.	50
4.5	Local frequencies in Y direction using WFR techniques. Scale is radians per pixel in y direction.	50
4.6	General schematic of edge connection problem.	52
4.7	Example Cost functions in deformation mode. It is shelf-like region throughout the probable solution in location where edges are detected (solid lines). On the other hand, dashed line is cost function for smooth region. The offset and standard deviation for dashed line is calculated from phase gradient $\omega_{x,y}$	53
5.1	Example of simulated data which will be processed with new proposed algorithm. Scale is in radians.	56
5.2	Local phase gradients in X direction	56
5.3	Local phase gradients in Y direction	57
5.4	Fourier correlation coefficient (FCC). The values are high in smooth regions of the interferogram. Along the fault, the values are smaller. Since noise is added in the simulated data, the maximum value of FCC is this map is not reaching 1 which is maximum allowed value of FCC.	57
5.5	Edges detected on FCC domain. The small artifacts are due to morphological operations performed on the edges for post processing.	58
5.6	Unwrapped solution. (a) is unwrapped by cost functions generated by SNAPHU itself.(b) is unwrapped with new cost functions. The big error at lobes is unavoidable since the signal is aliased in original simulated signal. It is observed that SNAPHU smoothens the unwrapped solution and underestimates the deformation.	59
5.7	True error in the unwrapped solution. Since the data is simulated, it is possible to get a quantitative estimate of errors.	59
5.8	An example of real life interferogram. The arrows show locations of 2 pixels which are used for testing of unwrapped result. There are 8 cycles difference between these two locations.	60

5.9	The unwrapped solution provided by SNAPHU. Branch cuts are also shown as black lines. As seen, SNAPHU is putting branch cuts across the smooth fringes.	61
5.10	The unwrapped solution provided by SNAPHU after setting up new cost functions. Branch cuts are also shown. As seen, now the SNAPHU is putting branch cuts which are not crossing smooth fringes. .	61
5.11	Example of cases where it is hard to detect edges since the fringe pattern remains smooth along the fault. Left images are simulated data and right images are their FCC. Since there is no clearly visible change in phase along the fault, the technique didn't work.	63

List of Tables

3.1	Setting up the cost functions for SNAPHU. Each arc has 4 parameters which defines the shape of these functions.	39
-----	---	----

Chapter 1

Introduction

This chapter will provide general introduction about motivation and research work presented in this document

Our environment is constantly changing and in the modern times, it's an absolute necessity to measure these changes, for the reasons of general welfare of human society or may be just out of curiosity. Making these observations is not always trivial, since they occur at different scales, different locations and are of different nature so specific techniques are required to monitor these changes. Remote sensing is the branch that measures these changes by collecting the information from a distance. Geodesy is the discipline that deals with measurements related to the Earth. Some specific branches of Geodesy are optical remote sensing, laser remote sensing, radar remote sensing etc. In the present context, radar remote sensing is under consideration. In this field, images are taken from a radar satellite revolving around the Earth. These images are then processed to extract required information.

1.1 Background

The Earth subsurface is constantly going through geological or other changes which sometimes cause the uplift or subsidence of the surface. Interferometric synthetic aperture radar (InSAR) is a specific application that deals with measurement of these deformations. This method is capable of measuring deformation at very large areal extents. Recent advances in the technique improved the accuracies up to sub-centimeter level. In this technique, two images are taken at different epochs and output of processing of these images is termed as an interferogram. This interferogram contains information about the deformation, topography, atmospheric signal along with noise. A typical example of an interferogram is shown in Fig. 1.1.

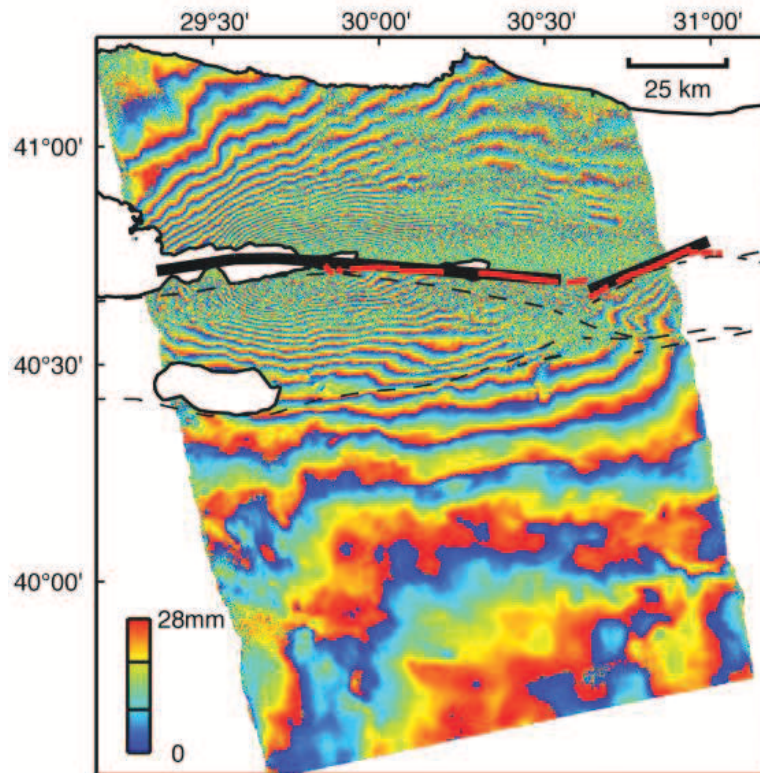


Figure 1.1: An example of an interferogram of Izmit earthquake occurred in 1999. The fault occurred during this earthquake is shown as a line. (Credits: NASA/University of Oxford)

From Fig. 1.1, the deformation of surface is visible in terms of fringes. One peculiar thing to notice is that both sides have different sequence of colors i.e. one side is red-to-blue and other side is blue-to-red as one moves away from the fault which is shown as thick black lines. This means that one side of the surface is going away from the radar satellite and other side is coming towards the satellite.

The magnitude of this deformation can be estimated by counting the number of fringes on both sides. One fringe corresponds to half a cycle of wavelength of the signal used to acquire the data as

$$\partial\theta = -\frac{4\pi}{\lambda}\partial\Delta R \quad (1.1)$$

where phase difference is $\partial\theta$ and path length difference is $\partial\Delta R$ (Hanssen, 2001). One cycle difference corresponds to 2π , which is equivalent to half the wavelength of the signal.

This process of counting fringes is phase unwrapping and this should be done automatically and mathematically. Phase unwrapping is very critical for deformation measurement since wrong unwrapping results will lead to wrong estimation of deformation. That is why phase unwrapping is a widely studied topic and is considered of utmost importance in InSAR field.

A measured phase value, by definition, is modulo 2π i.e. values are wrapped in an interval. Generally this interval is $(-\pi, \pi)$. This introduces ambiguity in the measurement. Phase unwrapping is the process of estimating this ambiguity. Phase unwrapping problem have been studied for more than 2 decades. Some of earliest works in this field were done by Hunt (1979) and Goldstein et al. (1988) which put the problem formulation into a mathematical perspective. Some of the other notable works in the field are done by Ghiglia and Romero (1989), (1994), (1996); Ghiglia and Pritt (1998); Zebker and Lu (1998); Chen and Zebker (2000a), (2001), (2002a), (2002b). General introduction to these formulations and algorithms are given in Chapter 2.

1.2 Problem formulation and research objective

Chen (2001) proposed an algorithm to solve phase unwrapping problem which is called SNAPHU. SNAPHU is a very efficient algorithm which formulates the phase unwrapping problem as a maximum a posteriori probability (MAP) estimation problem. MAP is a mode¹ of posterior distribution². So MAP provides a point estimate of an unobserved quantity given a certain event. This MAP problem is then transformed into cost functions and solved using network flow approaches. Details of this algorithm is provided in Chapter 3.

The cost functions are objective functions which upon optimization provide an unwrapped solution. But in regions where there is abrupt change in phase, or there are faults, the algorithm sometimes provides wrong solutions which can be identified by visual inspection (Fig. 5.9). Solving this problem becomes the objective of current research: Improving SNAPHU cost functions such that unwrapped solution improves in both faulty and smooth regions.

The research objective of the study is:

Improvement of statistical cost functions used in SNAPHU by incorporating additional parameters to improve phase unwrapping algorithm.

To fulfill this objective, research questions are formulated as:

Research question

How can the statistical cost functions used in SNAPHU be improved using additional parameters which gives an improved solution?

This question can then be further divided into two sub-questions:

Question I Can a faulty/non-faulty location be detected in an interferogram using edge and pattern recognition techniques?

¹mode of a probability distribution function is the value x at which the function takes maximum value.

²posterior distribution of a random event is conditional probability that is assigned once the event has taken place.

Question II How to incorporate this information in terms of cost function which can be used in SNAPHU to solve for the unwrapped interferogram?

These questions are addressed in subsequent sections of this report. Question I is addressed in detail in Chapter 4. Question II is discussed in brief in Chapter 3 and in detail in Chapter 4.

Fault detection is done using Windowed Fourier Transformation and edge detection techniques. At the same time, phase gradient (section: 2.2.2) information is also extracted using same technique. This information is then included in SNAPHU by making new cost functions. At the end of the report, results are shown and further recommendations are made.

1.3 Scope of study

The objective of current study is to develop a new unwrapping algorithm which provides better cost functions. These cost functions are then used as an input for SNAPHU which works as a solver routine. Since SNAPHU is being used as a solver, new cost functions are formulated in the same way as generated by SNAPHU itself. Working on improvement of SNAPHU solver routine is beyond the scope of current study.

SNAPHU deals with both topography and deformation. Current study is only focused on cost functions in deformation mode. These cost functions can be generated using different programming tools such C, C++ or MATLAB. MATLAB is used to implement new method of cost function generation.

1.4 Outline

This report is mainly divided in three major sections. First chapter deals with explanation of phase unwrapping algorithms and general philosophy behind unwrapping techniques. Simulated data examples are given wherever necessary. Sub-

sequent chapters deal with signal processing where fault detection algorithm is worked out. Final chapters include details about how the cost functions are incorporated in SNAPHU.

Initially the proposed algorithm is tested with simulated data only. Once the simulated data provided satisfactory results, a few real interferograms are unwrapped. In the case of simulated data, it is possible to provide a quantitative estimate of quality of unwrapped solution. On the other hand, real life data are checked visually and some manual calculation are made to check the performance. At the end of report conclusions are provided and recommendations are made.

Chapter 2

Phase Unwrapping Algorithms

Fundamental idea behind phase unwrapping is elaborated in the beginning. In later section, state-of-the art algorithms are explained.

2.1 Introduction

In active remote sensing, a signal is generated from a source and then the reflected or backscattered signal is collected at the receiver. Usually this signal is an oscillating wave with one or more frequencies. The phase of a signal is defined as fraction of a wave cycle that has elapsed relative to the origin. So the phase, by definition, is modulo 2π meaning the values are wrapped in an interval $(-\pi, \pi)$. This phase is termed as wrapped phase since integer number of cycles is missing from it. The wrapped phase is denoted as ϕ^w which is defined as:

$$\phi^w = W \{ \phi^a \} = \text{mod} \{ \phi^a + \pi, 2\pi \} - \pi, \text{ with} \quad (2.1)$$

$$\phi^a = 2\pi k + \phi_N \quad (2.2)$$

$$\phi_N \in [-\pi, \pi) \quad (2.3)$$

where W is wrapping operator, ϕ^a is the absolute phase of the signal, k is integer

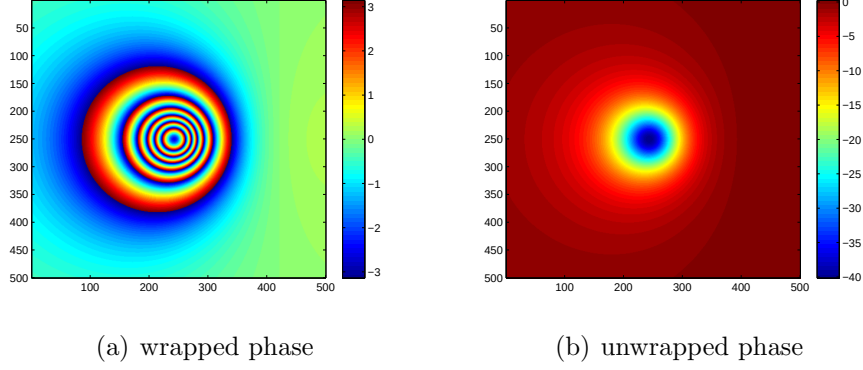


Figure 2.1: Example of a simulated wrapped and unwrapped phase field. The scale is in radians. Wrapped phase field is obtained by removing inter number of cycles from unwrapped phase.

ambiguity number and ϕ_N is additive phase noise (Hanssen, 2001).

The process of retrieving integer ambiguity k is phase unwrapping. In the most general and least restrictive sense, phase unwrapping is an impossible problem since integer number of cycles are missing from the phase (Ghiglia and Romero, 1996). Several methods have been developed for estimation of unambiguous phase which results into unwrapped phase (ϕ). So unwrapped phase (ϕ) is defined as the phase which is obtained by phase unwrapping process. It is a trivial procedure to wrap an absolute/unwrapped phase. On the contrary, unwrapping this wrapped field is of utmost importance, being one of the most significant sources of error (Chen, 2001). A better idea about wrapped and unwrapped phase field can be understood from Fig. 2.1.

Several algorithms and heuristics have been developed to solve the problem of phase unwrapping. The general approach and assumptions are elaborated in following section.

2.2 General method and assumption

The most important assumption that is taken for phase unwrapping problem formulation is that Nyquist sampling criterion is satisfied.

2.2.1 Nyquist criterion

The first and foremost assumption is that Nyquist¹ criterion is satisfied for most, though not necessarily complete, part of the phase field (Chen, 2001). This means that aliasing is avoided for most part of the signal space and *smoothness* is expected. Only at a physical discontinuity, eg. a fault due to earthquake, there is not sufficient sampling and the criterion is not satisfied.

This assumption becomes the foundation of the phase unwrapping problem. Under this assumption, it can be presumed that modulus of the wrapped phase difference between two neighboring pixels is not more than half a cycle (i.e. π) almost everywhere. Before going into the details of implications of Nyquist criterion, general unwrapping approach is discussed in following section.

2.2.2 Reconstruction of phase from phase gradients

The general approach for unwrapping process is to estimate phase gradients (differences) between two neighboring pixels and reconstruct the unwrapped phase from phase gradient field by integrating gradients over the complete phase field (Hunt, 1979). If the difference between two pixels is known, phase values can be obtained by adding the gradient to one of pixels to estimate the unwrapped values of another pixel. So phase unwrapping is generally done by first estimating these gradients.

Now the mathematical expression for phase gradients is derived. The coordinate convention that is used throughout this thesis is shown in Fig. 2.2.

The phase difference of wrapped phase ϕ^w array of size $M \times N$ at a random point on a grid (i, j) are defined in both x and y direction of 2-D phase field as:

¹Nyquist sampling: If the highest frequency in a signal is B, then a sampling rate of 2B sample per unit distance is sufficient to reconstruct the complete signal.

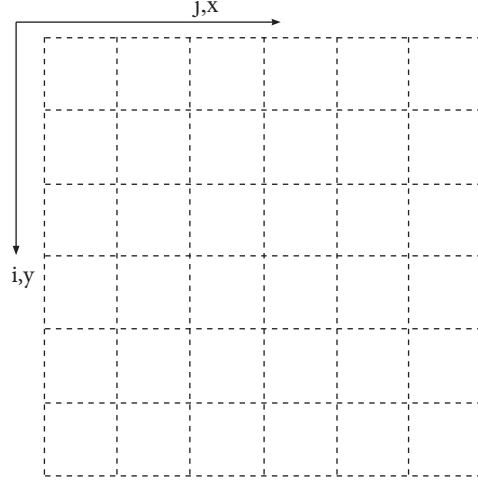


Figure 2.2: Coordinate convention followed in the current thesis. (i, j) are in accordance with standard matrix coordinate system. (x, y) are image coordinates.

$$\Delta\phi_{i,j}^{w,(y)} = \begin{cases} W\{\phi_{i+1,j}^w - \phi_{i,j}^w\} & i = 0, \dots, M-2; j = 0, \dots, N-1, \\ 0 & \text{otherwise} \end{cases} \quad (2.4)$$

$$\Delta\phi_{i,j}^{w,(x)} = \begin{cases} W\{\phi_{i,j+1}^w - \phi_{i,j}^w\} & i = 0, \dots, M-1; j = 0, \dots, N-2, \\ 0 & \text{otherwise} \end{cases} \quad (2.5)$$

where W is wrapping operator and ϕ^w is wrapped phase. Similarly phase gradients for any phase field can be calculated.

An example of 1-D unwrapping

A simple 1-D phase unwrapping example is given in Fig. 2.3 to get better idea about Nyquist criterion and phase unwrapping in general. The phase values are shown as a fraction of a cycle. On the arrows, the phase jump is shown. A peculiar situation arises at the location of the arrow where the modulus of phase jump is more than 0.5. Since we assume that Nyquist criterion is satisfied, this phase

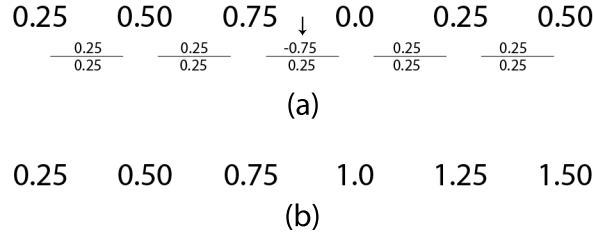


Figure 2.3: Example of 1-D wrapped and unwrapped array. (a) Wrapped phase values shown as a fraction of a cycle (b) Unwrapped phase values obtained under Nyquist criterion.

jump is offset to 0.25 from -0.75 by adding one cycle. Once the phase jumps are calculated, a simple integration is done by adding the phase jumps from left to right. The phase differences can be more than half a cycle in a real life situation and the assumption of Nyquist sampling does not hold any more. For example, if the phase jump is 1.25 instead of 0.25, a complete cycle will be missing in the unwrapped estimate. Moreover, since integration is done pixel-by-pixel, the error will propagate to the complete array.

There is nothing much that can be done in 1-D case, but in case of 2-D unwrapping, the gradient is in both x and y directions. Here, further mathematical tools can be used to mitigate this situation where error propagates to the complete phase field. The techniques are further explained in next section.

2.2.3 Conservative and non-conservative field

A vector field is said to be conservative if the integration from one point to another point is path independent. A conservative field is necessarily an irrotational field meaning the curl over a closed loop is vanishing. The phase field is a measured physical quantity which can be considered as a *potential* or *scalar* field (Chen, 2001). Mathematically, a scalar field ϕ is irrotational if curl of the field is equal to zero as:

$$\nabla \times \nabla \phi = 0 \quad (2.6)$$

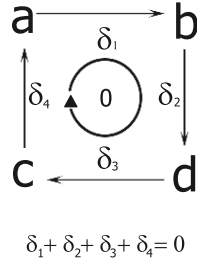


Figure 2.4: Irrotational field. The δ values are phase gradients, in clock-wise order. The summation of these gradients is equal to zero in an irrotational field.

where ∇ is gradient operator as:

$$\nabla = \left(\frac{\partial}{\partial x} \mathbf{i} + \frac{\partial}{\partial y} \mathbf{j} \right) \quad (2.7)$$

where $\mathbf{i}$ and $\mathbf{j}$ are unit vectors in x and y directions respectively.

True phase gradient $\nabla\phi$ is unknown and has to be estimated with wrapped gradient $\nabla\phi^w$ (Hanssen, 2001).

Eq. (2.4) can be further understood by Fig. 2.4. The values shown are phase values and closed loop integral is identically equal to zero. Mathematically, the evaluation of smoothness is given as:

$$r_{i,j} = \Delta\phi_{i,j}^{w,(x)} + \Delta\phi_{i,j+1}^{w,(y)} - \Delta\phi_{i+1,j}^{w,(x)} - \Delta\phi_{i,j}^{w,(y)} \quad i = 0, \dots, M-1 \quad j = 0, \dots, N-1 \quad (2.8)$$

where r is termed as a *residue* and its values can be 0, -1 or 1 (Goldstein et al., 1988).

The observed signal is not necessarily conservative throughout the phase field. This is caused by noise and undersampling. If the condition of irrotational field doesn't hold, the integration becomes path dependent and unwrapped phase is not

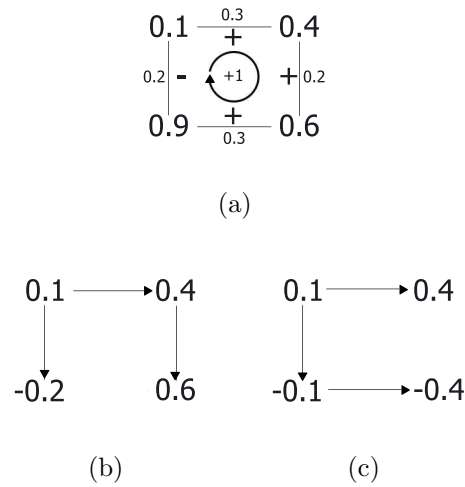


Figure 2.5: Path dependence in irrotational field. (a) The 2x2 pixel phase field is not conservative since the sum of residues does not vanish. (b) and (c) are unwrapped solutions, but integration is done by different paths. As observed, the final unwrapped result is not the same and depends on choice of path.

unique any more. Fig. 2.5 further shows how the solution is path dependent.

Goldstein et al. (1988) indicated in his work that if unequal number of positive and negative gradients occurs in a loop (in Fig. 2.4 the cycle is a loop), the residues do not vanish. This problem must be dealt with since only one true unwrapped solution is possible. To deal with path dependence, the residues are discharged by connecting them making *cuts* that serve as barriers for integration process. Significant work has been done during last couple of decades on phase unwrapping problem in InSAR. Following section provides details about some noteworthy works.

2.3 Previous formulations and Algorithms

Several attempts have been made to solve the problem of phase unwrapping in various fields eg. SAR interferometry (InSAR)(Goldstein et al. (1988); Ghiglia and Pritt (1998); Costantini (1996); Chen (2001)), MRI scanning (Szumowski

et al. (1994); Moon-Ho Song et al. (1995); Ching et al. (1992)) etc. In this section, some major and widely accepted methods for SAR interferometry are explained. One common attribute to all the algorithms is that phase gradients are estimated rather than direct unwrapped phase values. These gradients are then integrated to calculate unwrapped phase values.

These mathematical formulations and algorithms can mainly be divided into three sections:

- Branch cut algorithms (2.3.1)
- L^p norm framework (2.3.2)
- Network flow algorithms (2.3.3)

2.3.1 Branch-cut algorithm

Branch-cut method of phase unwrapping was first introduced by Goldstein et al. (1988). This algorithm identifies residues and connects them using *branch-cuts*. By Eq. (2.8), if the value of a residue (r) is $+1$, it is considered as positively charged, and if it is -1 , it is considered as negatively charged. The branch-cuts connect positive and negative charges in such a way that a cut is 'charge free'. These cuts serve as a *fence* for the integration. This way, the inconsistency of the solution is avoided and a final unique unwrapped field is achieved independent of integration path. Fig. 2.6 shows the location of a branch cut. As observed, a branch-cut is a cut drawn from one residue to another via shortest path. The goal is to minimize the total length of the cuts since these are the only locations where a phase discontinuity (a jump of more than half a cycle) is allowed. Shortest path ensures that the number of these locations is kept to a minimum which is purely a mathematical assumption and no real world physical interpretation is made.

In the original paper, Goldstein et al. (1988) proposed a selection of a random residue and then a 3x3 search box is created to look for further residues. If a residue is found, it is connected to an existing residue via a cut. If the cut is neutral, it is considered complete and next residue is selected and same steps are repeated. If a residue is not found, the size of the box is increased to 5x5 and same

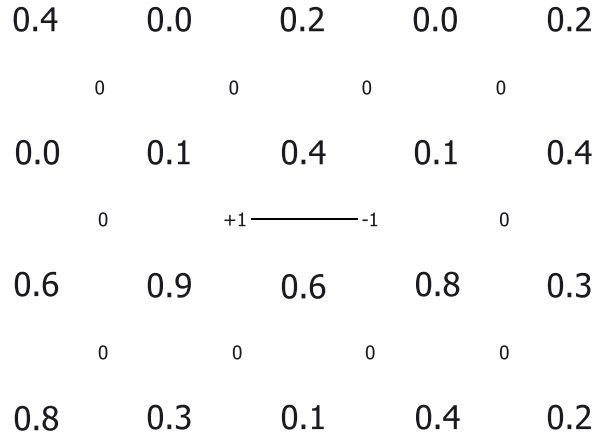


Figure 2.6: Example of a branch-cut. The phase values are shown as a fraction of a cycle. Residues are mostly 0. Positive and negative residues are connected by a line. This line serves as barrier and integration is not allowed over this cut.

steps are taken. In a nutshell, all residues are connected producing charge-free cuts, hence the solution is consistent, and probable locations of the discontinuities are estimated. Since integration is not allowed over a cut, these are the only locations where a phase jump of more than half a cycle is allowed. Finally, to minimize the length of the cuts, if boundary of the field is encountered, the tree is connected to the boundary and deemed neutral, considering the boundary to be superconducting.

This algorithm had several shortcomings. The most prominent one was that the residues are not checked as visited/non-visited. So the same residues are visited multiple number of times. Also, in areas of high noise (dense residues), the cuts have dendritic appearance, and hence are called *trees*. These dense cuts may isolate some parts of the field, resulting into holes in the unwrapped solution (Zebker and Lu, 1998).

Following the same philosophy of neutralizing residues, minimum spanning tree

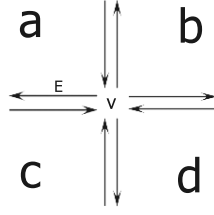


Figure 2.7: Nodes and arcs representation of a phase field. a, b, c, d are pixels. E is an arc and V is node. A complete phase field can be represented in terms of nodes and arcs accordingly.

approach is developed which is described in following section. Earlier attempt was made by Ching et al. (1992) for MRI data which was adapted by Chen (2001) for InSAR phase unwrapping.

2.3.1.1 Minimum Spanning Tree

An improved version of branch-cut algorithm is developed by (Chen, 2001) which uses Minimum Spanning Tree (MST) approach. This problem is well studied in graph theory. A graph is an abstract representation of a set of objects where some pairs of the objects are connected by links. The interconnected objects are represented by mathematical abstractions called vertices, and the links that connect some pairs of vertices are called arcs² (Trudeau, 1993). Reader is directed to Ahuja et al. (1993) for detailed study of the problem. Only general outlines are provided here.

A phase field is defined in terms of nodes and arcs where an arc is the edge that divides two neighbouring pixels and node is intersection point of arcs. Fig. 2.7 shows this representation.

These nodes and arcs are represented as a connected, weighted, undirected graph $G = (V, E)$, where V is set of all the nodes and E is set of all the arcs. A function

²It should be noted that arcs are generally termed as edges in graph theory, but since edge is used in a different sense in later sections, the connection between vertices is termed as arc here.

$w : E \rightarrow \mathbb{R}$ assigns weight $w(e)$ to each arc e . The minimum spanning tree T of graph G is achieved by minimizing:

$$w(T) = \sum_{e \in T} w(e) \quad (2.9)$$

Prim's³ is used to solve for T . Chen (2001) proposed Algorithm 1 to solve for a minimum spanning tree problem.

Algorithm 1 MST algorithm pseudo code

Start

Choose one charge as root of tree T

while T doesn't not contain all charges **do**

Start

Find the shortest Euclidean path λ from any node on T to a charge not on T incorporate λ into T , and set arc distance λ to zero

End

end while

integrate phase

End

Once this tree is generated, the branches are considered as branch-cuts. Unwrapping is then done by flood-fill⁴ method (Goldstein et al., 1988). One major advantage of this method is that it generates tree, which is, by definition, loop-free. So it is guaranteed that the final unwrapped solution will be complete without holes.

2.3.2 L^p norm optimization

Another method of phase unwrapping is formulating the problem as a polynomial optimization problem. This method was proposed by Ghiglia and Romero (1996). Some attempts were made earlier as L^2 norm (least square) optimization (Takajo

³Prim's algorithm is a greedy algorithm which is used to find a minimum spanning tree of network. Reader is directed to (Chen, 2001) or (Ahuja et al., 1993) for further details.

⁴flood-fill integration method finds connected pixels without a branch-cut along the arcs between them and unwrapping is done under the assumption that Nyquist sampling criterion is fulfilled.

and Takahashi (1988); Ghiglia and Romero (1994)). L^p norm optimization approach was a more general method which combined the earlier methods. General discussion of the approach is given here.

In this approach, the objective is minimizing the difference between wrapped phase differences ($\Delta\phi^w$) and unwrapped phase differences ($\Delta\phi$) ((2.5)) under L^p norm approach. For an $M \times N$ grid from Eq. (2.5), general L^p norm optimization objective is defined as:

$$\begin{aligned} \text{minimize} \{ & \sum_{i=0}^{M-2} \sum_{j=0}^{N-1} w_{i,j}^{(y)} |\phi_{i+1,j} - \phi_{i,j} - \Delta\phi_{i,j}^{w,(y)}|^p \\ & + \sum_{i=0}^{M-1} \sum_{j=0}^{N-2} w_{i,j}^{(x)} |\phi_{i,j+1} - \phi_{i,j} - \Delta\phi_{i,j}^{w,(x)}|^p \} \quad (2.10) \end{aligned}$$

where w is the weight, indices x and y represent gradients in respective directions and p is the scale of optimization.

It is to be noted that Eq. (2.10) is a generic optimization criterion which is independent of the solving method. It is possible that same optimization criterion is used for guiding the unwrapping process but different methods are used for optimization process. This may result into completely different solutions (Chen, 2001).

A particular case of $p = 2$ was solved by Ghiglia and Romero (1994) using fast transformation (*discrete cosine transformation*). Only after this, a general L^p norm optimization was proposed.

In L^p norm optimization, no gradients are explicitly dropped, so usually the unwrapped solution is smooth and generally not congruent. A non-congruent solution means that the difference between wrapped and unwrapped gradients is not always integer number of cycles. Though some attempts were made again to enforce congruence after optimization (Ghiglia and Pritt, 1998), but this is done at the expense of L^p norm estimate. In any case, the congruence is not ensured before

the optimization and is not a part of the solving methods.

Ghiglia and Romero (1994) derived relationship of L^p norm optimization and previous branch-cut algorithm. They concluded that for $p = 0$, Eq. (2.10), gives a value which is equal to sum of number of samples for which wrapped and unwrapped gradients are not equal. So L^0 case minimizes the sum of cut lengths which is, in fact, equivalent to branch-cut/minimum spanning tree method of unwrapping.

These developments relied upon one fundamental idea: defining the problem as an optimization problem. Chen (2001) further explored the idea and took the optimization objective beyond polynomial optimization and created a more general optimization objective which is discussed later.

2.3.3 Network-flow algorithms

L^p norm optimization approach put the unwrapping problem in a mathematical perspective, but it didn't suggest any method to solve the problem. Network flow theory overcomes this issues. Network flow theories are widely studied in computer science and one major advantage of these algorithms is that along with optimization objectives, they provide means of solving the objective functions.

Idea of network-flow optimization is not new to SAR interferometry. Some implicit network-optimization models were given in 1990's (Pritt, 1996) but first explicit attempt was made by Costantini (1996, 1998). In this model, a grid of each 2x2 pixels is represented as a node and phase gradients are represented as arcs⁵ (Fig. 2.7). Using the notation of branch-cut method, a positive residue is denoted as surplus and a negative residue is denoted as demand. General minimum cost flow problem is formulated in the following section.

⁵An edge and arc are different naming conventions of same quantities. Arc is a commonly used term in Network theory and edge is more common term in Graph theory. For the sake of simplicity, edges in graph theory are referred as arcs.

2.3.3.1 Minimum-cost flow optimization

A brief mathematical model is described in this section. Consider same graph $G(V, E)$ again with source (supply) $s \in V$ (+1 residue nodes) and sink (demand) $t \in V$ (-1 residue nodes) where arc $(p, q) \in E$ has capacity⁶ $k(p, q) > 0$, and flow⁷ $\chi(u, v) \geq 0$ and cost⁸ $C(u, v)$, which is constant in present case. Say total supply is A (sum of positive residues) and total demand is B (sum of negative of residues). The minimum-cost flow problem is formulated as:

$$\text{minimize } \left\{ \sum_E C(u, v) \cdot \chi(u, v) \right\} \quad (2.11)$$

under the constraints

$$\chi(u, v) \leq k(u, v) \quad (2.12)$$

$$\chi(u, v) = -\chi(v, u) \quad (2.13)$$

$$\sum_{w \in V} \chi(u, w) = 0 \text{ for all } u \neq s, t \quad (2.14)$$

$$A = B \quad (2.15)$$

In words, flow is taken from supply nodes to demand nodes along the arcs on a network such that total cost is minimum. A robust algorithm is developed for the solution of minimum cost flow problem by Goldberg (1997). Costantini (1998) also concluded that Eq. (2.11) is equivalent to Eq. (2.10) when $p = 1$. So minimum cost flow is a specific case of L^p norm optimization where $p = 1$. This method not only provides a mathematical formulation, but a method to solve makes it a practical approach.

⁶capacity: maximum allowed flow through an arcs

⁷flow: amount of flow through an arc

⁸cost: cost of unit flow through an arc

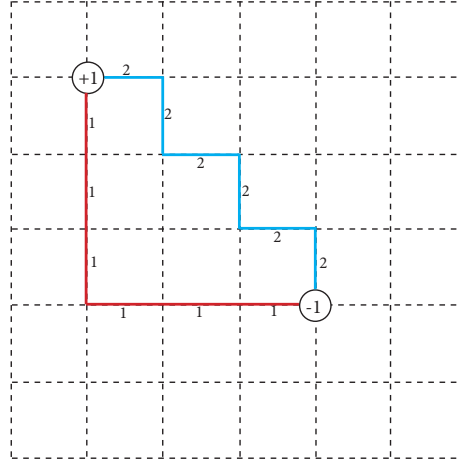


Figure 2.8: Example of an MST and MCF solution. Blue line is an MST solution where the total cost of the flow is 12. Red line is one of the probable solutions given by MCF optimization where total cost of flow is 6.

Comparison between MCF and MST

If all the costs along the arcs on a grid are same, minimum cost flow (MCF) and minimum spanning tree (MST) produce same results. They differ only if the costs are different. So assigning cost is a way of guiding the flow which ultimately becomes the branch cuts. An example of how different costs produce different results is shown in Fig. 2.8. The number written on an arc is an assigned cost (C) of flow associated with that particular arc. Total flow (χ) on the network is one, since there is only one supply node (A) and one demand node (B) with unit flow generation at supply node.

Blues lines (*zig-zag structure*) is a possible solution based on MST. The total cost of flow is 12 in this case. The red line (2 long straight lines connecting residues) is a possible MCF solution where the total cost of flow is 6. Thus, different cost will force the flow through different arcs which will result into different unwrapped solution.

MCF and MST generally provides different results. In some cases, MST is a

better solution and sometimes MCF gives a better solution. If proper weights are assigned, MCF is more likely to produce better results since more reliable gradients are given higher weight. Nevertheless, both approaches are purely mathematical and no relation to physical reality is incorporated in the formulation and solution of the problem.

2.3.3.2 SNAPHU

SNAPHU is another network flow optimization based algorithm. Here a brief introduction is given. Chapter 3 provides details about this algorithm. Chen (2001) incorporated network-flow optimization with more general cost functions. Rather than using only polynomial optimization objective (Eq. (2.10)), he used more general optimization objective functions. He formulated the phase unwrapping problem as a maximum *a posteriori* probability (MAP) estimation problem. Additionally, congruence is incorporated within the algorithm, which was not the case in L^p norm optimization.

The use of SNAPHU in current context is very important since external cost functions can be generated and then can be given as an input to SNAPHU. So application specific new cost functions can be generated and optimized with different settings.

Chapter 3

Statistical Models and Non-linear Optimization

Chen and Zebker (2000b) proposed a phase unwrapping algorithm based on statistical models and non-linear optimization¹. In this approach, the phase unwrapping is considered as a maximum *a posteriori* probability estimation (MAP) problem. MAP is a Bayesian statistical approach to calculate a point estimate of a posterior probability distribution where posterior distribution is a conditional probability which is assigned after relevant evidence is taken into account (Dekking, 2005). This approach assigns maximum likelihood value to a random variable based on empirical data. For a probability distribution function (PDF) of the vector of observables $\underline{\alpha} \in \mathbb{R}^m$ which is parameterized as $f_{\underline{\alpha}}(\underline{\alpha}|\beta)$, with $\beta \in \mathbb{R}^n$ unknown. The maximum likelihood value of β is defined as

$$\hat{\underline{\beta}} = \arg \max_{\beta \in \mathbb{R}^n} f_{\underline{\alpha}}(\underline{\alpha}|\beta). \quad (3.1)$$

A model of the statistical characteristics of the interferogram phase and its relation to the quantities such as image intensity or image coherence was developed.

¹Non-linear optimization is the process of solving a system of equalities and inequalities, collectively termed constraints, over a set of unknown real variables, along with an objective function to be maximized or minimized, where some of the constraints or the objective function are nonlinear (Bertsekas, 1998)

Given a 2-D wrapped phase field Φ^w ², the minimization of this objective function results into $\hat{\Phi}$, which is an estimate of absolute phase Φ^a , where $\hat{\Phi}$ maximizes the total conditional probability density functions (PDF) $f(\Phi|\Phi^w)$. By definition, MAP distribution function can be any function which makes the problem set up very flexible.

MAP puts the unwrapping problem in terms of a more general mathematical optimization objective. Then MAP distribution functions are mapped to cost functions which are then used to solve the problem over a discrete wrapped phase field. Using this statistical model, non-linear network flow optimization problem is solved to find the unwrapped solution. The critical information developed in the complete algorithm is the development of the objective function, which assigns appropriate value to a probable solution. The more robust these objective functions are, the better the final solution is. Of course, development of these functions depends greatly on the probabilistic relation of physical quantities to the mathematical models.

3.1 MAP distribution functions

MAP distribution functions are objective functions that allow a greater flexibility in creating the optimization objective. Instead of dealing with phase values directly, the unwrapped phase derivatives $\Delta\Phi$ are estimated, given wrapped phase derivative $\Delta\Phi^w$ (Chen, 2001). Further assumption is taken that the PDF $f(\Phi|\Phi^w)$ is separable, meaning individual unwrapped gradients $\Delta\phi$ are statistically independent from each other given their wrapped counterparts $\Delta\phi^w$. This assumption of independence is, in fact, not strictly correct though (Chen, 2001). This is to be noted that upper and lower case letters are not random variables but their specific instances. $f(-)$ represents a random variable eg. $f(\Phi|\Phi^w)$ is equivalent to $f_{\Phi|\Phi^w}(\Phi|\Phi^w)$.

A general unwrapped PDF is shown in Fig. 3.1. This illustration represents the

²Capital letters denote complete array and small letters denote individual instances.

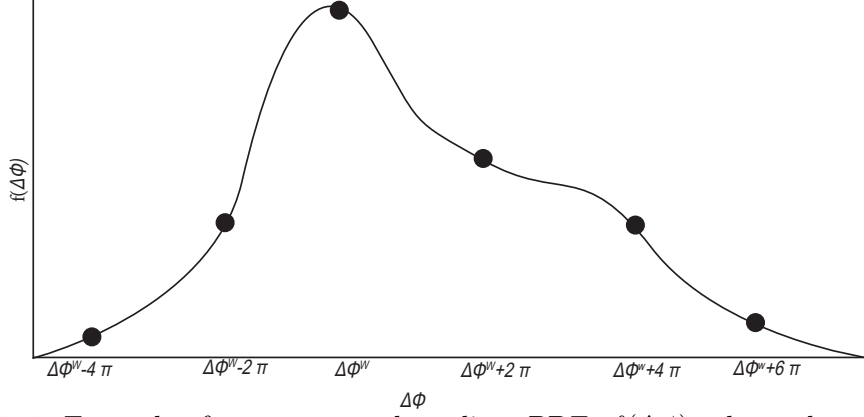


Figure 3.1: Example of an unwrapped gradient PDF. $f(\Delta\phi)$ takes only values which are integer offset from wrapped gradient $\Delta\phi^w$ under the assumption of congruence. The shape of PDF is general and doesn't represent the real shape of MAP PDFs used in SNAPHU (Chen, 2001).

general idea behind the PDFs. Random variable $f(\Delta\phi)$ takes values at points where the unwrapped solution is integer number of cycles offset from the wrapped gradients ($\Delta\phi^w$).

Now the objective of the optimization problem is defined as:

$$f(\Delta\Phi|\Delta\Phi^w) = \text{maximize} \left\{ \prod_{\kappa} f(\Delta\phi|\Delta\phi^w) \right\} \quad (3.2)$$

κ represents that multiplication is taken over all the row-wise and column-wise gradients. $f(\Delta\phi|\Delta\phi^w)$ is an individual instance of a MAP PDF.

Eq. (3.2) is a maximization problem. Generally, it is more convenient to deal with minimization problems. Also, multiplication is a costly operation (Chen, 2001). So the objective is transformed into:

$$\text{minimize} \left\{ - \sum_{\kappa} \log(f(\Delta\phi|\Delta\phi^w)) \right\} \quad (3.3)$$

Thus, MAP distribution functions are negative logarithms of the unwrapped gra-

dient PDFs:

$$g_{\kappa}(\Delta\phi_{\kappa}, \Delta\phi_{\kappa}^w) = - \sum_{\kappa} \log(f(\Delta\phi_{\kappa}|\Delta\phi_{\kappa}^w)) \quad (3.4)$$

where g_{κ} is a single MAP PDF.

The unwrapped values follow the assumption of congruence, i.e. the unwrapped gradient is offset to wrapped gradient by only integer number of cycles. So g is a probability mass function rather than a density function. $f(\Delta\phi|\Delta\phi^w)$ is a conditional probability mass distribution, which can be further simplified as:

$$f(\Delta\phi|\Delta\phi^w) = \begin{cases} \frac{f_{\Delta\phi}(\Delta\phi)}{\sum_{l=-\infty}^{\infty} f_{\Delta\phi}(\Delta\phi^w + 2l\pi)} & \text{if } \Delta\phi = \Delta\phi^w + 2k\pi, \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

Where k and l are integers. The denominator is independent of $\Delta\phi$ so has no effect on minimization problem of Eq. (3.3). So Eq. (3.5) can be written as:

$$g_{\kappa}(\Delta\phi_{\kappa}, \Delta\phi_{\kappa}^w) = -\log(f(\Delta\phi_{\kappa})) \text{ for } \Delta\phi_{\kappa} \equiv \Delta\phi^w + 2k\pi \quad (3.6)$$

The Eq. (3.6) is the MAP distribution function. It depends on $f(\Delta\phi)$ so a good knowledge of $f(\Delta\phi)$ is important.

From Eq. (3.6), each arc has a PDF assigned to it. These PDFs are discrete functions at integer offset values. It means that different between an unwrapped phase and its wrapped counterpart is integer number of cycles. For a MAP estimate, solution benefits from inclusion of additional information. Intensity I and coherence γ are available for all the SAR interferograms. So Eq. (3.6) becomes

$$g_{\kappa}(\Delta\phi, \Delta\phi^w) = -\log(f(\Delta\phi|I, \gamma)) \text{ for } \Delta\phi_k \equiv \Delta\phi^w + 2k\pi \quad (3.7)$$

where I is average intensity of SAR images used to create the interferogram and γ is the magnitude of interferogram complex correlation coefficient. The local coherence is the cross-correlation coefficient of SAR image pair estimated over a few pixels in both range and azimuth directions. Coherence γ is calculated as:

$$\gamma = \frac{E \{y_1 y_2^*\}}{\sqrt{E \{|y_1|^2\} E \{|y_2|^2\}}}} \quad (3.8)$$

where y_1 and y_2 are complex values of pixels of SAR images and E is expected value of a function (Hanssen, 2001).

It should be noted that the shape of MAP PDF varies throughout the interferogram, since each phase gradient has a different I and γ values.

Theoretically, a MAP PDF can be any function. In SNAPHU, these PDFs are Gaussian or Gaussian-like functions. Gaussian-like functions are combination of a Gaussian function joined with a shelf-like region. Also it should be noted that logarithm of a Gaussian function is a parabolic function.

SNAPHU formulated the phase unwrapping problem as MAP estimation problem. MAP PDFs described here provides mathematical basis of the optimization problem. Once the problem is set up, SNAPHU solved the MAP PDFs using cost function approach which are described in following section.

3.2 Deformation and MAP cost functions

One of the many prime interests of SAR interferometry is study of deformation. Deformation is displacement of ground or objects on ground between two different epochs. These time instances are acquisition times of two SAR images which are used to create an interferogram. Deformation can be caused by several phenomena such as earthquake, volcanic activity, glacial activities, subsidence, etc. It is assumed that the signal acquired is deformation signal along with noise i.e.

$$\phi_{signal} = \phi_{defo} + \phi_{noise} \quad (3.9)$$

$$\Delta\phi_{signal} = \Delta\phi_{defo} + \Delta\phi_{noise}. \quad (3.10)$$

Under the assumption that only deformation signal is present in an interferogram, signal is independent of intensity I . Further, the functions behave similar in both range and azimuth directions since there is no physical difference in terms of de-

formation in both directions (Chen, 2001). So Eq. (3.7) becomes:

$$g_{\kappa}(\Delta\phi, \Delta\phi^w) = -\log(f(\Delta\phi_{\kappa}|\gamma)) \text{ for } \Delta\phi_{\kappa} \equiv \Delta\phi^w + 2k\pi. \quad (3.11)$$

So, from Eq. (3.11), deformation mode cost functions are only functions of γ . Chen (2001) assumed a direction correlation between coherence γ and deformation which is a weak assumptions. It was assumed that coherence is low at the locations of phase discontinuities which is true but low coherence can be caused by several other reasons eg. vegetation or water bodies. SNAPHU estimates phase jumps at these location which sometimes result into wrong unwrapped solution which can be visually inspected. This is the reason behind current research work: to find a better physical quantity which reflects the probability of a branch cut and then use that information to create new cost functions for better unwrapped solution. Now, the statistics for deformation model is developed. Important point to be noted here is that MAP functions are assigned values empirically and certain parameters are adjusted by analyzing the unwrapped results.

First, a threshold coherence value, γ_{min} is selected. Above this value of coherence, cost functions are parabolic, negative logarithms of Gaussian noise PDF, centered at zero (dashed line in Fig. 3.2). For the locations where coherence is less than selected γ_{min} , the PDF is defined in 3 different regions which is combination of parabolic and shelf-like regions (solid line in Fig. 3.2).

$$g(\Delta\phi) = \frac{\Delta\phi^2}{\sigma_{\Delta\phi}^2} \text{ for } \gamma \geq \gamma_{min} \quad (3.12)$$

In the situations where γ is less than γ_{min} , the cost function is:

$$g(\Delta\phi) = \begin{cases} \frac{\Delta\phi^2}{\sigma_{\Delta\phi}^2} & \text{if } |\Delta\phi| \leq \Delta\phi_{crit} \\ g_d & \text{if } \Delta\phi_{crit} < |\Delta\phi| \leq \Delta\phi_{max} \\ \frac{(|\Delta\phi| - \Delta\phi_{max})^2}{C_{tail}\sigma_{\Delta\phi}^2} & \text{if } |\Delta\phi| > \Delta\phi_{max} \end{cases} \quad (3.13)$$

Where

$$|\Delta\phi_{crit}| = \sqrt{g_d\sigma_{\Delta\phi}^2} \quad (3.14)$$

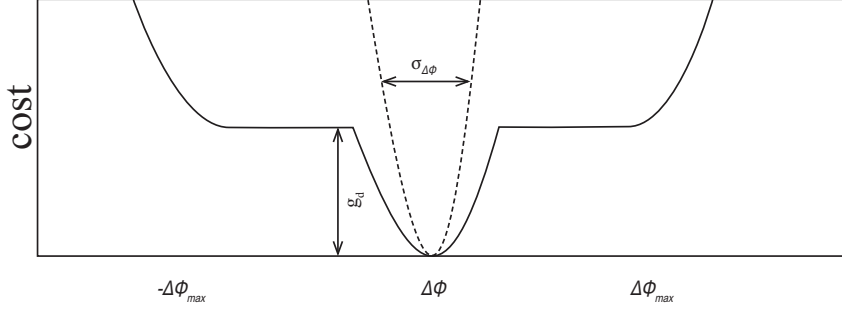


Figure 3.2: The MAP cost functions in both cases where $\gamma \geq \gamma_{min}$ and $\gamma \leq \gamma_{min}$. Solid line is for the case where $\gamma < \gamma_{min}$. Otherwise its a dashed line (Chen, 2001).

Here, C_{tail} is a constant. γ_{min} , g_d and $\Delta\phi_{max}$ are assigned values empirically.

3.3 SNAPHU solver routine

SNAPHU uses network-flow optimization approach to solve for the unwrapped phase. A brief description is already provided in Chapter 2. Here, SNAPHU algorithm is described in a bit more detail. The reader is referred to Ahuja et al. (1993) and Chen (2001) for further explanation.

Within SNAPHU, the first solution is produced using either L^0 (minimum spanning tree, section: 2.3.1.1) or L^1 (minimum cost flow, section: 2.3.3.1) optimization instead of directly solving with cost functions. L^0 and L^1 optimizations are explained in Chapter 2. This improves the efficiency of the problem since the first solution is fast to obtain. Only after that, iteratively a *Dynamic cost cycle canceling* algorithm is used to further improve the quality of solution to estimate the maximum probable solution.

SNAPHU uses this approach of first estimation to make the solver computationally efficient. The L^0 approach works well in high coherence areas where residues are not dense. But in residue rich areas, the algorithm doesn't provide a good solution

with respect to minimum-cost flow approach. It should be noted that the network is originally optimized with respect to L^0 metric, and then it is possible to improve this network with respect to minimum-cost flow approach that is, in fact, L^1 metric based. Detailed analysis of cycle-canceling is out of current context. Reader is suggested to go through Ahuja et al. (1993) and Chen and Zebker (2000a) for more elaboration. As evident from the name itself, cost-cycle canceling is based on cycles (i.e. loops) iteratively augmenting flow in closed loops, and adjusting cost of flow.

The use of SNAPHU in present research work is as a tool that maximizes the total conditional probability density function i.e. $f(\Phi|\Phi^w)$. MST and MCF are only starting points of the solution which gives a quick first estimate. Then the problem ultimately converges towards the final objective. So the final solution can, in fact, be completely different than MST or MCF. Also, final solution depends on the first estimates. Strikingly different solutions are produced depending on first solution being MST or MCF based.

3.4 Analysis of unwrapping algorithm using SNAPHU

The purpose of current study is to improve SNAPHU. But first, it needs to be established that improved cost functions is one of the ways to achieve better solution. Methodology adopted for testing is to create simulated data and then unwrap it with different settings in SNAPHU. Then create new cost functions, and test the effect of these functions on unwrapping algorithms.

SNAPHU has a versatile solver routine. It solves the networks using MST, MCF and cost-cycle canceling algorithms. Additional flags can be passed to stop the solver at only MST and MCF solution without going into cost-cycle canceling. The capability of inputting external cost functions enables a user to test the unwrapping algorithm's sensitivity towards different costs. This feature is exploited in detail in following section with descriptions about how cost functions are gen-

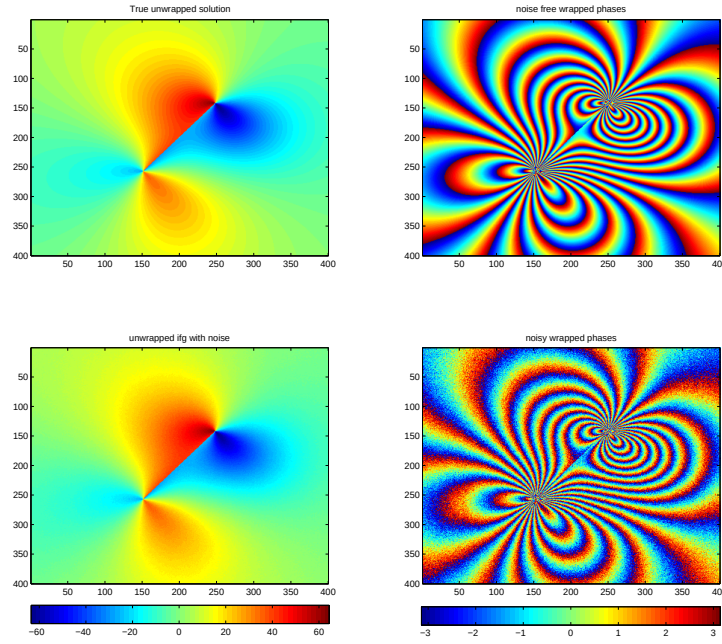


Figure 3.3: Example of simulated data. (i) Top left is unwrapped phase (ii) Top right is wrapped phase (iii) Bottom left is same unwrapped phase but 30° standard deviation noise is also added (iv) similarly wrapped phase with noise. Scale is in radians.

erated and input in SNAPHU. A few examples are shown below.

Example 1 First example is a deformation pattern caused by a fault running through an area. This deformation pattern is shown in Fig. 3.3.

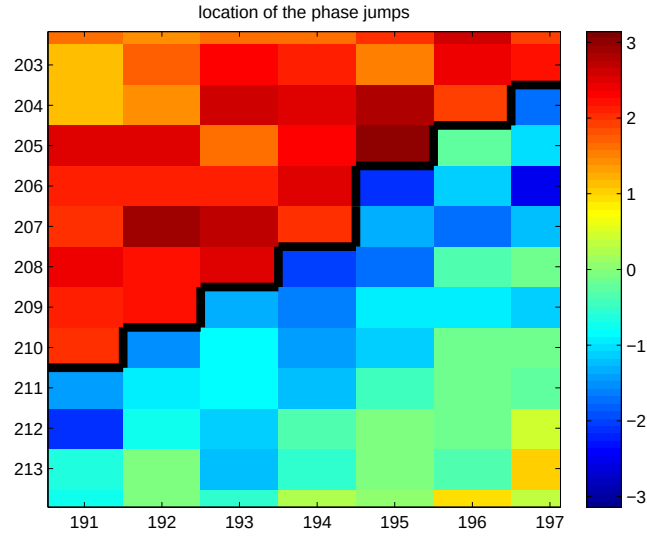


Figure 3.4: Phase jumps in the simulated data. These are the locations where phase gradient is more than half a cycle. These locations are obtained from true unwrapped solution and are shown on a wrapped solution.

Since the data is simulated, it is possible to know the locations of the phase jumps. From a true unwrapped solution, the locations are identified where the difference between two pixels is more than half a cycle. Examples of these jumps are shown in Fig. 3.4 (page: 34) where the image is zoomed in to make these locations visible.

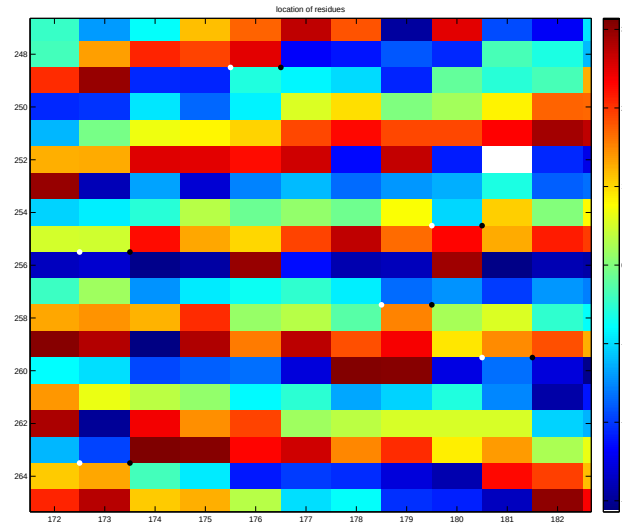


Figure 3.5: Residues in the simulated data are shown as black and white dots. Black dots correspond to -1 and white dots correspond to $+1$ residue. These are the locations where closed loop integral of phase gradients is not vanishing. It is observed that residues generally occur in pairs.

Residues are important information to observe. The concept of residues is explained in Fig. 2.4 (page: 14). Example of residues is shown in Fig. 3.5. Here also, only a small part of the interferogram is shown. In the Fig. 3.5, the white dots are positive residues and black dots are negative residues. Generally when residues occur due to noise, they occur in pairs. So most likely, the branch cut will directly connect them along the arc between the two residues. It is observed that in cases where there is a phase jump because of real discontinuity eg. fault due to earthquake, residues don't occur in pairs. Connecting them is a major issue that makes the unwrapping problems difficult to address.

Now, this simulated unwrapped phase is wrapped and then unwrapped with SNAPHU. The minimal SNAPHU configuration file is shown in Appendix A. For detailed settings, reader is referred to (SNAPHU configuration file). The main point to be noted in this file is that no input cost file is defined. So cost functions are created by SNAPHU itself. It is possible to set own functions which are discussed later in this example.

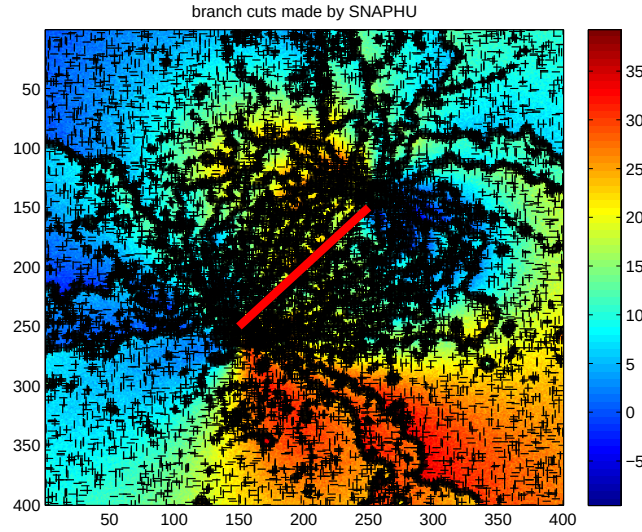


Figure 3.6: Branch cuts made by SNAPHU. Dark red line is the location of the real fault where there are true phase jumps. Black lines are branch cuts estimated by SNAPHU. It can be observed that SNAPHU branch cuts don't represent the real location of the discontinuity.

This unwrapped result is shown in Fig. 3.6. The real location of the fault is shown as a dark red line. The black lines are the lines where SNAPHU predicted the branch cuts. It can be observed that the branch cuts estimated by SNAPHU don't coincide with real phase jumps which resulted into a wrong unwrapped solution.

The errors are shown in Fig. 3.7 and Fig. 3.8. Scale is number of cycles.

From Fig. 3.7, it can be seen that there are large zones of error in unwrapped solution which is a direct consequence of mislocating the branch cuts. One point to be mentioned here is that no coherence information is additionally supplied for unwrapping. But as elaborated in previous chapter, its not a very reliable source of information. The numbers of pixels with error values are shown in Fig. 3.8 as a bar graph. It is observed that about 8% of the total pixels are wrongly unwrapped.

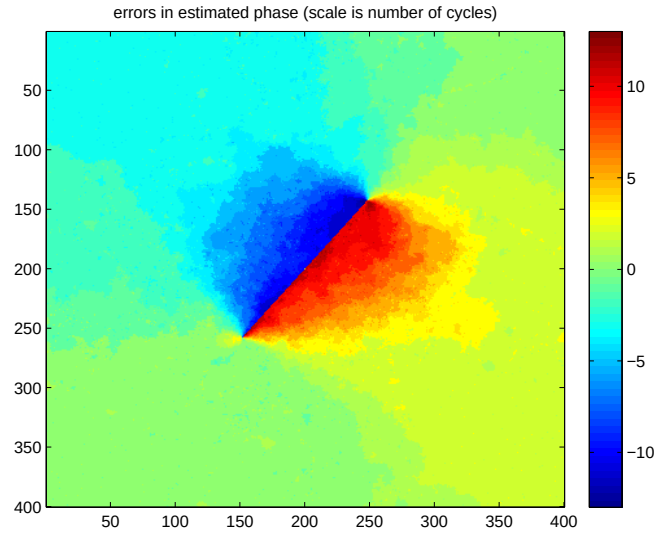


Figure 3.7: Error of the unwrapped solution generated by SNAPHU with respect to true unwrapped solution. Scale is number of cycles.

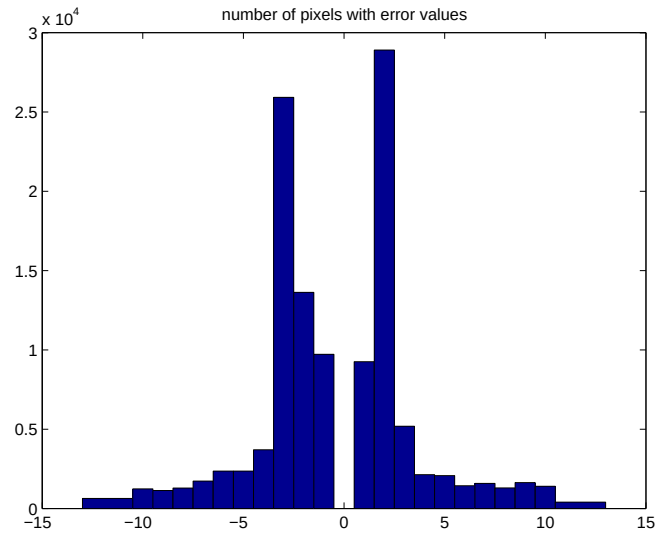


Figure 3.8: Number of pixels with error values. x -axis is number of cycles.

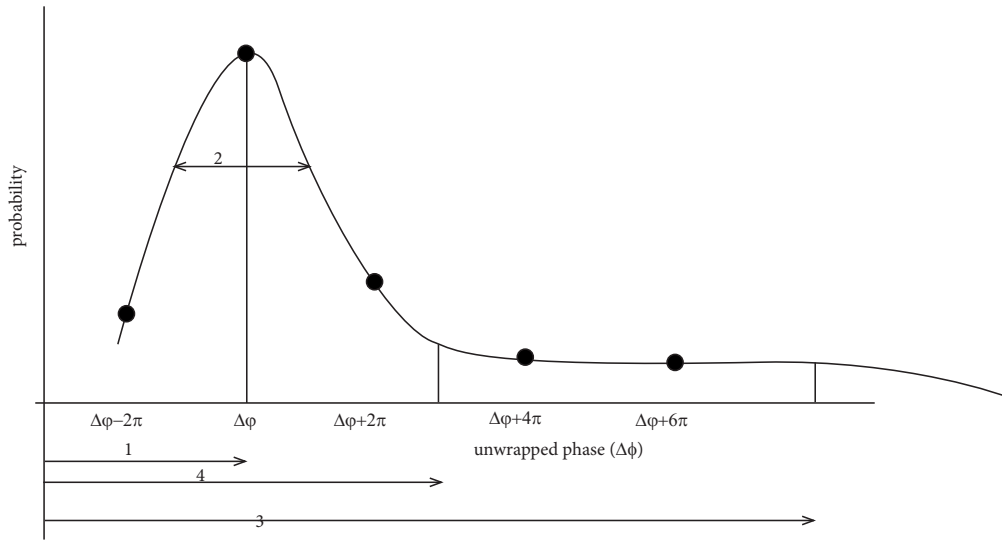


Figure 3.9: Example of cost function parameters. The figure is not to scale and provides a general idea about the shape of the objective functions. Number 1-4 represent the columns as shown in Table: 3.1

The wrong unwrapped solution provided by SNAPHU is direct consequence of wrong estimation of cost functions. It is possible to generate external cost functions and then optimize them using SNAPHU. Since the true locations of phase jumps are known already from the simulated unwrapped solution, new cost functions are generated using this information. From Fig. 3.9, there are 4 parameters that are used to set up these cost functions. So for each arc, there are 4 parameters describing the shape of the function.

The 4 parameters shown in Fig. 3.9 are explained in Table: 3.1.

The location of the phase jumps are known using simulated true unwrapped solution. The cost functions are set in such a way that there is maximum probability of having a branch-cut at locations of true phase jumps. A SNAPHU configuration file is changed and input external costfile is set provided.

Now, again SNAPHU is used to find unwrapped solution. The resulted unwrapped

First column	mean value of the Gaussian cost function, the location where peak is occurring
Second column	sigsq, variance of the Gaussian part of the cost functions
Third column	values at which cost soars proportional to the square of the values exceeding this value
Fourth column	this value is called laycost in SNAPHU, values where shelf like region starts

Table 3.1: Setting up the cost functions for SNAPHU. Each arc has 4 parameters which defines the shape of these functions.

solution is shown in Fig. 3.10.

The error plot of the new unwrapped solution is shown below.

From Fig. 3.11 and Fig. 3.12, the significance of the cost functions can be understood. Improved cost functions provide better results. Initially, about 8% of pixels were wrongly unwrapped when no cost functions were provided to SNAPHU. But when better cost functions are generated using true simulated data, only about 0.1% is pixels are wrongly unwrapped.

Example 1 establishes that results produced by SNAPU greatly depend on cost functions. In this example, new cost functions are generated using simulated data where locations of phase jumps were known. Since in real like interferograms, true locations of phase jumps are not known, the location of phase jumps needs to be estimated using some other technique.

The technique adopted for current study is Fourier analysis which is explained in next chapter.

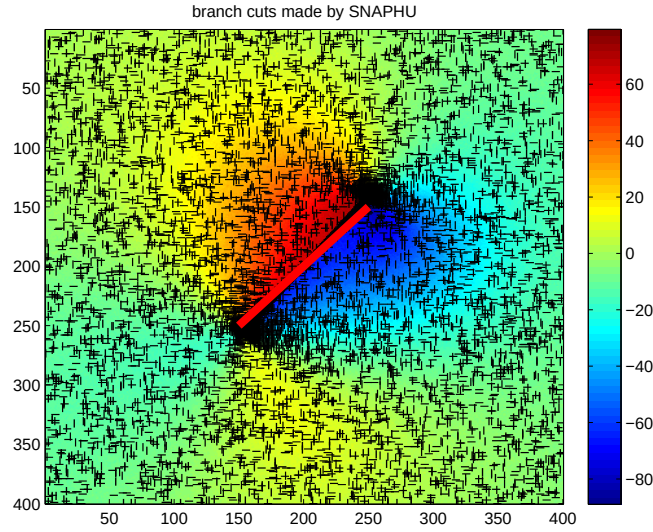


Figure 3.10: The unwrapped solution generated by SNAPHU using external cost functions which are generated using true simulated unwrapped solution. Since the locations of true phase jumps are known, function is set such that it is more probable to have a branch-cut at the location of true phase jumps. Using this information, cost functions are generated which provides improved results.

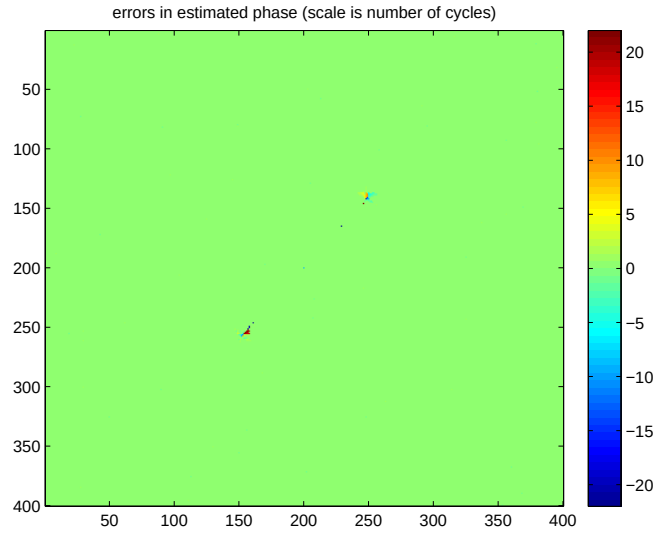


Figure 3.11: The error of the unwrapped solution using externally created cost functions.

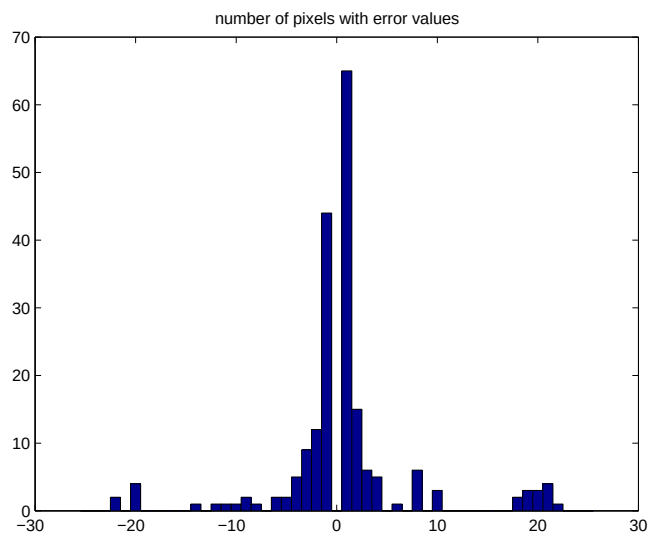


Figure 3.12: The number of pixels with error values of the unwrapped solution using manually created cost functions.

Chapter 4

Edge detection and Cost function generation

Edge detection is a widely studied topic in various fields. Edges can be loosely defined as locations where a certain property of an image changes eg. brightness, coherence etc. Canny edge detector is one the most used method of edge detection. Reader is directed to Canny (1986) for details about the technique.

For the purpose of current study, edges are defined as locations of phase discontinuity. Generally, phase varies smoothly throughout a coherent interferogram. At a location of a physical discontinuity, the phase changes abruptly. These locations are critical in terms of phase unwrapping problem. A branch-cut should be put at these locations to avoid integration since integration is done under Nyquist sampling criterion and non branch-cut locations are smooth. But before edge detection technique is applied, the interferogram needs to be transformed into phase gradient map. This is done by Windowed Fourier Transformation (WFT). WFT is referred as band-limited or time limited Fourier transformation also. The techniques are quite old but using this technique for phase pattern analysis to locate faults is rather new. Phase gradients ($\Delta\phi$, Eq: (2.5)) is calculated by computing difference between two neighboring pixels. These measurements include noise which makes a wrong estimate of the gradients. A better estimate of phase gradient can be obtained by including the neighboring pixels and then using this to find the locations

of discontinuity. Fourier analysis technique is suitable here since it is less sensitive to noise. Qian et al. (2005) and Kemaio (2007) proposed WFT which is described below.

4.1 Windowed Fourier Transformation

WFT is a tool to analyze local frequencies in an interferogram. These frequencies are visible in terms of fringe pattern. Further, fringe frequencies are directly proportional to phase gradients. At regions where there is high fringe rate directly infers high phase gradients. Generally, the pattern is expected to change smoothly throughout an interferogram. Now, the mathematical development of WFT is elaborated as shown by Kemaio (2007).

A general complex interferometric phase pattern can be written as:

$$\phi_c(i, j) = \exp(\iota \cdot \phi^w(i, j)) \quad (4.1)$$

where $\iota = \sqrt{-1}$, $\phi_c(i, j)$ is complex phase at a point (i, j) on $M \times N$ grid and $\phi^w(i, j)$ is wrapped phase at same location. Local frequencies (instantaneous frequencies) are expressed as:

$$\omega_x(x, y) = \frac{\partial \phi^w(x, y)}{\partial x} \quad (4.2)$$

$$\omega_y(x, y) = \frac{\partial \phi^w(x, y)}{\partial y} \quad (4.3)$$

where $\omega_x(x, y)$ and $\omega_y(x, y)$ are local phase gradients. Phase distribution in a window around a pixel (i, j) can be estimated as:

$$\phi^w(x, y) \approx \omega_x(i, j)(x - j) + \omega_y(i, j)(y - i) + \phi^w(i, j). \quad (4.4)$$

The transformation equations of WFT are given as:

$$\mathcal{F} = \mathcal{F}(x, y, j, i) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \phi_c(x, y) h_{j,i,\xi,\eta}^* dx dy \quad (4.5)$$

$$\phi_c(x, y) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathcal{F}(j, i, \xi, \eta) h_{j,i,\xi,\eta} d\xi d\eta dj di \quad (4.6)$$

where $\mathcal{F}$ is Fourier spectrum of phase ϕ_c , h is Fourier basis function, $*$ represents complex conjugate of a function. The windowed Fourier element h is:

$$h_{j,i,\xi,\eta} = h(x - j, y - i) \exp(\iota \xi x + \iota \eta y) \quad (4.7)$$

The Fourier basis function is infinite in spatial extent. On the other hand, the WFT basis function is limited because of limited size of the window function. So $\mathcal{F}$ provides local frequencies which are not possible to see in normal Fourier spectrum. The transformation is called Gabor Transformation if $h(x, y)$ is a Gaussian window (Kemaio, 2007) as:

$$h(x, y) = \exp(-x^2/2\sigma_x^2 - y^2/2\sigma_y^2) \quad (4.8)$$

where σ_x and σ_y are standard deviation of Gaussian function in x and y directions respectively. The Gaussian window is normalized such that $\|h(x, y)\|_2 = 1$.

On the basis of this transformation method, two techniques are developed:

- *Windowed Fourier filtering*
- *Windowed Fourier ridges*

4.1.1 Windowed Fourier filtering(WFF)

In this technique, the Fourier spectrum is limited and then inverse WFT is performed. The threshold (thr) is set by user. The equations for the transformation are shown next.

$$\bar{\mathcal{F}}(x, y) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\eta_1}^{\eta_h} \int_{-\xi_1}^{\xi_h} \bar{\mathcal{F}}(\xi, \eta, j, i) h_{j,i,\xi,\eta} d\xi d\eta dj di \quad (4.9)$$

where

$$\bar{\mathcal{F}}(j, i, \xi, \eta) = \begin{cases} \mathcal{F} & \text{if } |\mathcal{F}(j, i, \xi, \eta)| \geq thr, \\ 0 & \text{if } |\mathcal{F}(j, i, \xi, \eta)| < thr, \end{cases} \quad (4.10)$$

where $\bar{\mathcal{F}}$ denoted the threshold spectrum. It should be noted that the frequencies in the Eq. (4.9) are not $(-\infty, \infty)$. Instead, they are $[-\xi_1, \xi_h]$ and $[-\eta_1, \eta_h]$ for x and y directions respectively. So the spectrum $[\xi, \eta] \in [\xi_1, \xi_h] \times [\eta_1, \eta_h]$. This is done to save the processing time if the maximum and minimum frequencies are known. Also, $[\xi_1, \xi_h] \times [\eta_1, \eta_2] \subseteq [-\pi, \pi] \times [-\pi, \pi]$, since the Fourier basis functions $\exp(i\xi x + i\eta y)$ is cyclic and repeats outside this interval.

For example, the window size around a pixel (i, j) is 10×10 is chosen. Its Fourier spectrum is infinite in spatial extent. But the window size of h , the windowing function is chosen 3 times that of the spectrum window, since a $3\sigma_x$ and $3\sigma_y$ will contain 99.5 of the total energy.

Windowed Fourier Filtering (WFF) step iteratively convolves the complex wrapped phase map with different window functions with different frequencies. A smooth pattern in the phase map should have a high correlation with one of these windows. The magnitude of this correlation values is termed as Fourier Correlation Coefficient (FCC). Since it is a correlation value, the interval of values is $(0, 1)$. At the location of phase discontinuities, FCC values are low.

Example 1 The wrapped phase simulated data is shown in Fig. 4.1.

Now the WFT technique is implemented on this data. The window size is 10×10 in both x and y directions. The values of $\xi \in [-2, 2]$ and $\eta \in [-2, 2]$ with a jump of 0.1 and threshold is set to 3. The final resulted FCC is shown in Fig. 4.2. The ends of the faults also have very low values of FCC. The signal is sub sampled here and resolution is low to cover high deformation rate in these regions. The prime location of interest is low values at the fault. If there is noisy signal in some part of the interferogram, it will also have low values. Noisy here refers to the fact that there is no smooth pattern in deformation so no fringes. From the analysis of the FCC map, the values are high in the regions where there is

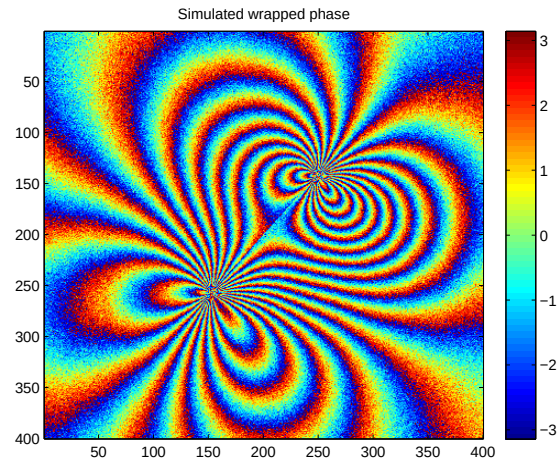


Figure 4.1: Example of simulated wrapped phase. Size is 400×400 . Area has a fault running across it. Rest of the interferogram is smooth. 20° standard deviation noise is added.

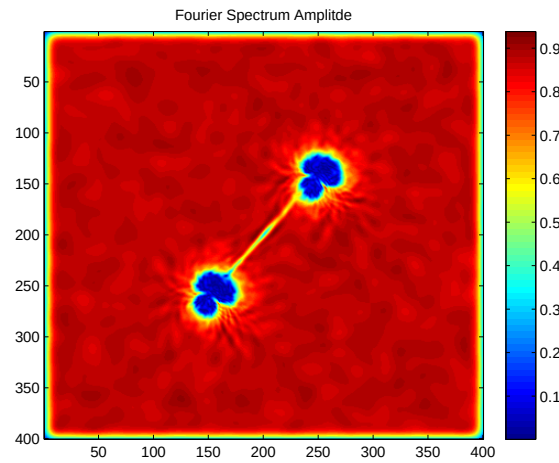


Figure 4.2: The Fourier Correlation Coefficient map of Fig. 4.1. Scale is $(0, 1)$. The regions with the fault have smaller values and regions with a pattern have higher.

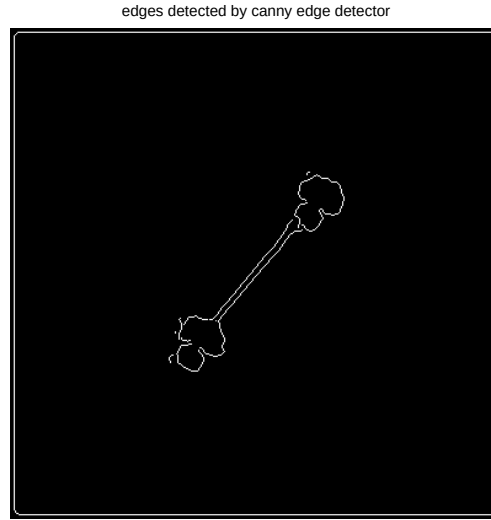


Figure 4.3: Edges detected by Canny Edge detector in Fig. 4.2. The amplitude is low in a band near the fault so 2 lines are detected. Post processing is required to finally make one edges out of 2 lines. The corners of the interferograms are also detecting edges because of insufficient sampling. These edges need to be dropped.

a smooth deformation signal. On the other hand, near the fault, the values are comparatively smaller. This is critical information in terms of phase unwrapping. If the signal is smooth, there should be less probability of having a branch cut and vice-versa. This information can be used to assist in setting up better cost functions.

To detect edges in this MAP, standard edge detection techniques can be used. For the current analysis, Canny Edge Detector was used. Detailed description of the technique is provided by Canny (1986).

In Fig. 4.3, the edges detected in FCC are shown. Further processing is required to use these edges which are addressed in later part of this chapter.

4.1.2 Windowed Fourier Ridges

Window Fourier Ridges (WFR) is another technique based on WFT. First the mathematical development is shown. Then the interpretation is given with examples. Consider a pixel (i, j) in the interferogram. The fringe pattern in the neighborhood of this pixel is most similar to one of the WFT elements with a particular ξ and η . This WFT element is referred as a ridge (R). The values of these ξ and η are taken as local frequencies at (i, j) . Since the similarity can be shown as amplitude of the spectrum, the local frequencies can be found out by:

$$[\omega_x(i, j), \omega_y(i, j)] = \arg \max_{\xi, \eta} |\mathcal{F}(i, j, \xi, \eta)| \quad (4.11)$$

So the local frequencies take the values of WFT elements frequencies. Now ridge and phase values are:

$$R(i, j) = |\mathcal{F}[i, j, \omega_x, \omega_y]|, \quad (4.12)$$

$$\theta(j, i) = \text{angle}\{\mathcal{F}[i, j, \omega_x, \omega_y]\} + \omega_x(i, j)j + \omega_y(i, j)i. \quad (4.13)$$

In WFR techniques, IWFT is not required. The values of ω_x and ω_y can then be used to find phase discontinuities. For most part of the interferogram, fringes change smoothly. Its only near the faults where there is an abrupt change in local phase gradients. To extract edges, again, standard edge detection techniques are used. In this case, Sobel Edge detector is used. Sobel provides location where gradient is more than set threshold. Local frequencies are shown in the Fig. 4.4 and Fig. 4.5.

Frequencies ω_x and ω_y are equal to phase gradients. This can be then used in formulation of cost function since they provide a better estimate than previous phase gradient concept introduced in SNAPHU. The new function based on phase gradient is formulated in next chapter.

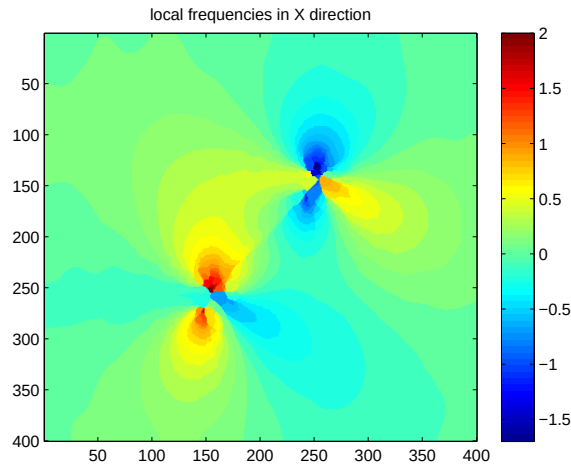


Figure 4.4: Local frequencies in X direction using WFR technique. Scale is radians per pixel in x direction.

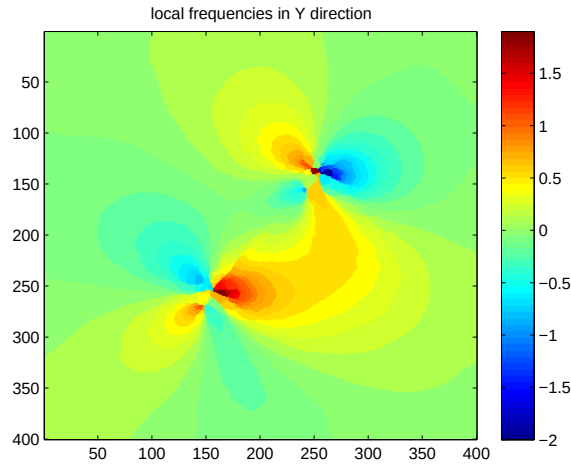


Figure 4.5: Local frequencies in Y direction using WFR techniques. Scale is radians per pixel in y direction.

4.2 Edge detection and processing

The edges detected have certain artifacts. For example, in Fig. 4.3, there are two lines. Similarly in the edges detected by local phase gradients, there are certain errors. These errors are mainly disconnected lines. The reason can be inaccurate thresholding or actual signal. This needs to be handled since the branch cuts should not be put across a discontinuity.

This requires feature extraction from the image (edges in current case). Snapping of the objects and healing the edges is an extensively studied topic particularly by GIS specialists (personal talks with Dr. Hugo Ledoux, OTB research institute, TU Delft). Estimating linear features is the first step. Edge detectors provide zigzag looking edges since phase discontinuities are not detected precisely. A general schematic of edge connection algorithm is shown in Fig. 4.6.

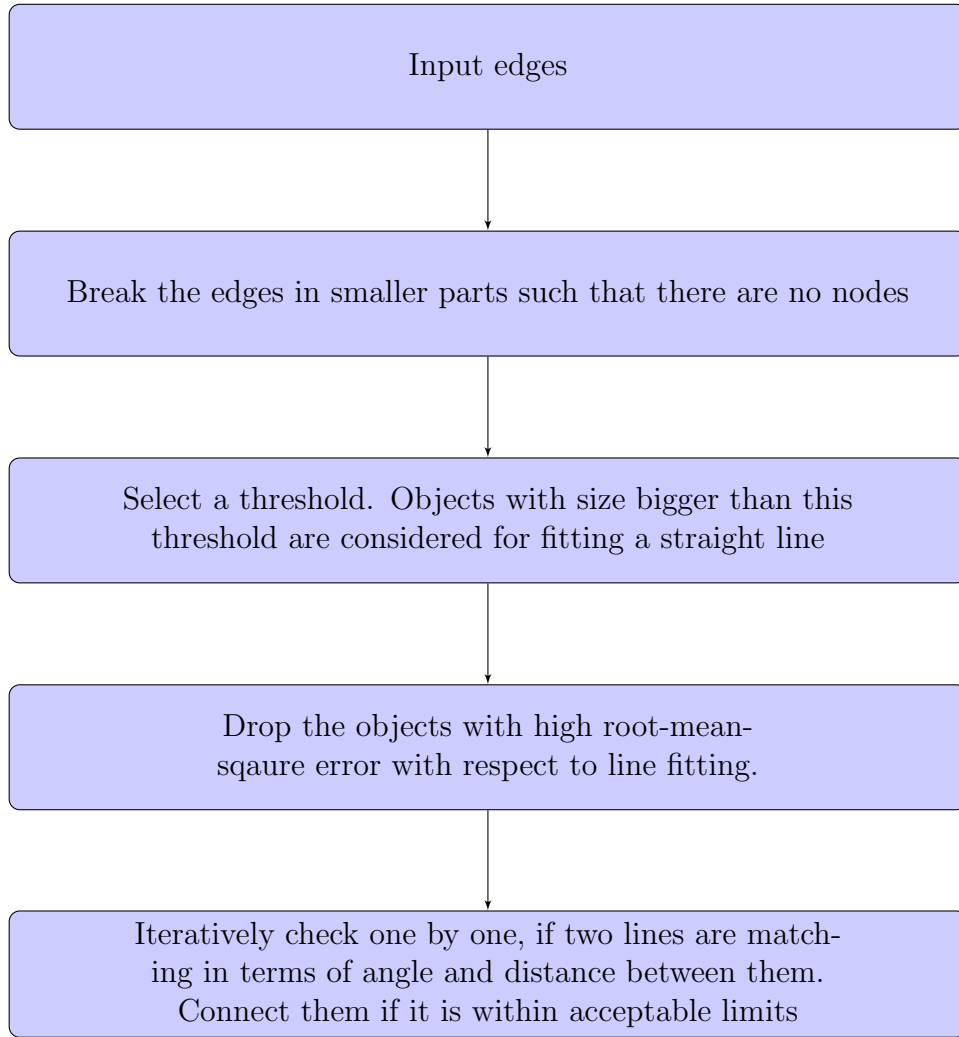


Figure 4.6: General schematic of edge connection problem.

4.3 New cost functions

After exploring the possible parameters which are better indicative of phase discontinuities, new cost functions need to be set up. To reiterate previous cost functions for deformation mode in SNAPHU from Eq. (3.7), page: 28,

$$g_{\kappa}(\Delta\phi, \Delta\phi^w) = -\log(f(\Delta\phi|I, \gamma)) \text{ for } \Delta\phi_{\kappa} \equiv \Delta\phi^w + 2k\pi$$

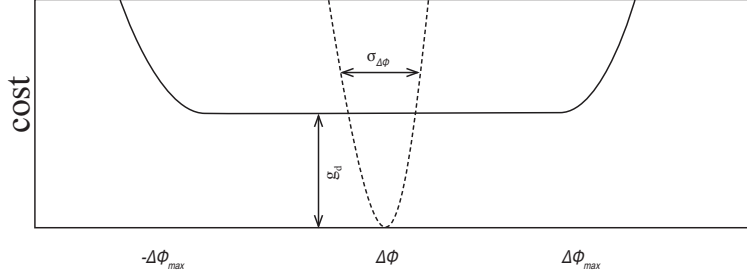


Figure 4.7: Example Cost functions in deformation mode. It is shelf-like region throughout the probable solution in location where edges are detected (solid lines). On the other hand, dashed line is cost function for smooth region. The offset and standard deviation for dashed line is calculated from phase gradient $\omega_{x,y}$

Following the same philosophy, new MAP PDFs are generated using estimated phase gradients and FCC. Mathematically, they can be defined as:

$$g_{\kappa}(\Delta\phi, \Delta\phi^w) = -\log(f(\Delta\phi)|FCC, \omega_{x,y}) \text{ for } \Delta\phi_{\kappa} \equiv \Delta\phi^w + 2k\pi \quad (4.14)$$

where κ represents PDF for one arc and k is integer. Eq. (4.14) creates new optimization objective functions. These functions behave different in different locations. The region where edges are detected, it is more likely that branch cut should be put. Since the number of offset cycles is unknown, all the possible values are equally likely. Hence, the function is flat shelf-like region as shown in Fig. 4.7. On the other hand, the locations where phase gradient is smooth, it is less likely to have a phase jump. Hence the shape of the PDF at these locations is a parabolic function, centered at wrapped phase gradient. Standard deviation is empirically adjusted using the FCC since FCC indicates the quality of the estimated gradients.

The PDF of Eq. (4.14) is then transformed into cost functions as explained in section 3.2. Once new cost functions are generated, SNAPHU can be used to find the unwrapped solution.

Chapter 5

Results and discussions

After development of new cost functions, it is required to test if the new cost functions are providing satisfactory results or not. Testing is done in two stages. First stage is testing with simulated data and then real data are tested in second stage.

First, the simulated data example is given in following section.

5.1 Testing with simulated data

An example of simulated wrapped phase is shown in Fig. 5.1.

First the signal processing step is done to calculate Fourier Correlation Coefficient (FCC; section: 4.1.1, page: 45) and estimated local phase gradients $\omega_{x,y}$. The plot is shown in Fig. 5.2 and Fig. 5.3.

The FCC plot is shown in Fig. 5.4. The values of FCC is high in the areas where there is a smooth pattern in the interferogram. Along the fault across the interferogram, the values are rather small.

Now, the edge detection is applied to both of them separately. There isn't enough gradient in $\omega_{x,y}$ domain. But edges in FCC domain seem reasonable. The edges are shown in Fig. 5.5. These edges have certain artifacts due to edge correction

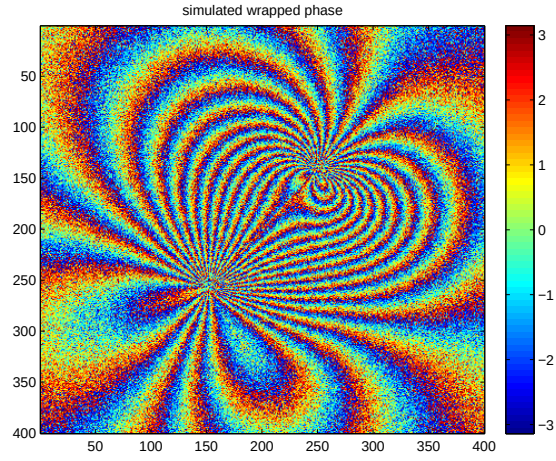


Figure 5.1: Example of simulated data which will be processed with new proposed algorithm. Scale is in radians.

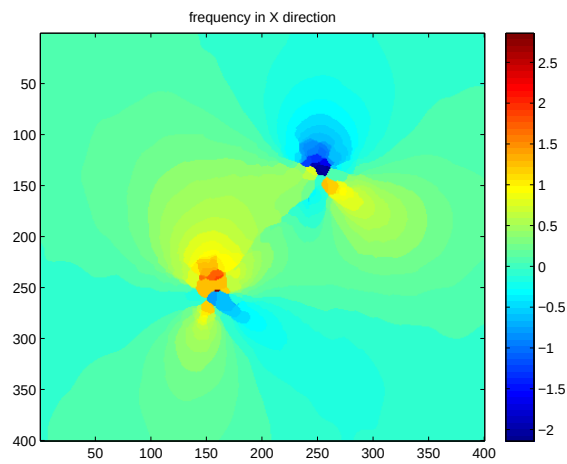


Figure 5.2: Local phase gradients in X direction

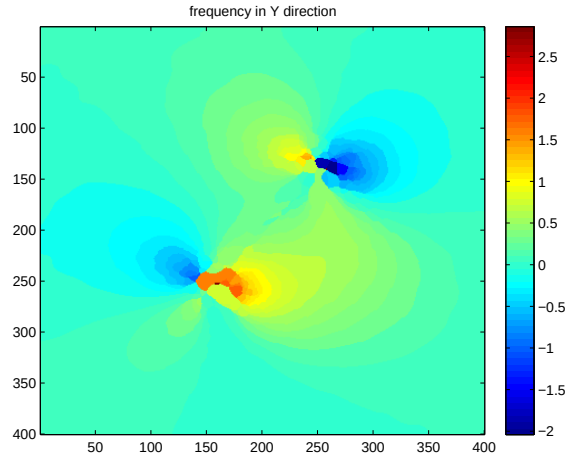


Figure 5.3: Local phase gradients in Y direction

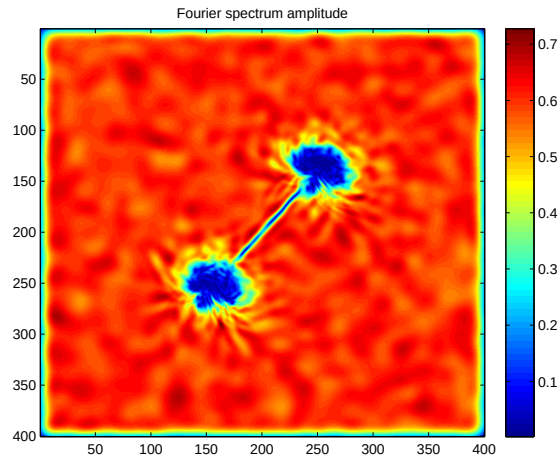


Figure 5.4: Fourier correlation coefficient (FCC). The values are high in smooth regions of the interferogram. Along the fault, the values are smaller. Since noise is added in the simulated data, the maximum value of FCC in this map is not reaching 1 which is maximum allowed value of FCC.

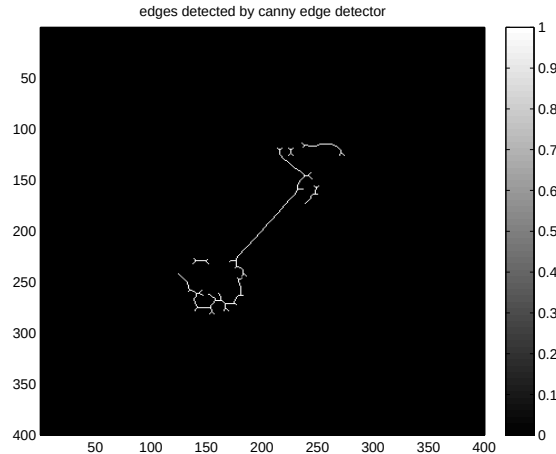


Figure 5.5: Edges detected on FCC domain. The small artifacts are due to morphological operations performed on the edges for post processing.

techniques applied. Morphological operations¹ are used to post process the image for edge corrections.

The new cost functions are set up as mentioned in previous chapter. Now phase unwrapping is done using these new cost functions. Result is shown in Fig. 5.6. It is observed that results are improved over the case where cost functions are generated by SNAPHU itself. Also, SNAPHU is producing outputs with lower estimation of deformation.

The error plot of the unwrapped solution is given in Fig. 5.7. About 2% of the pixels are wrongly unwrapped.

5.2 Testing with real data

Now one example of a real interferogram phase unwrapping is given. The size of the interferogram chosen is comparatively smaller than real life interferograms. The results are tested by visual inspection and some manual calculations.

¹Morphological operations are theory and techniques in digital image processing based on set theory, topology etc. Reader is referred to Serra (1982) for further reading about these operations.

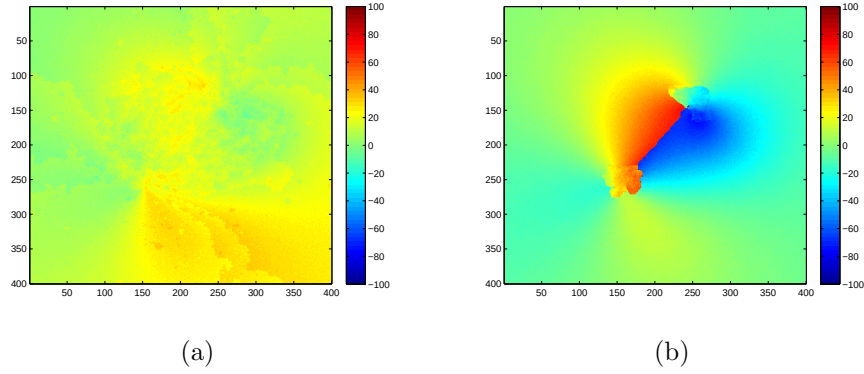


Figure 5.6: Unwrapped solution. (a) is unwrapped by cost functions generated by SNAPHU itself.(b) is unwrapped with new cost functions. The big error at lobes is unavoidable since the signal is aliased in original simulated signal. It is observed that SNAPHU smoothens the unwrapped solution and underestimates the deformation.

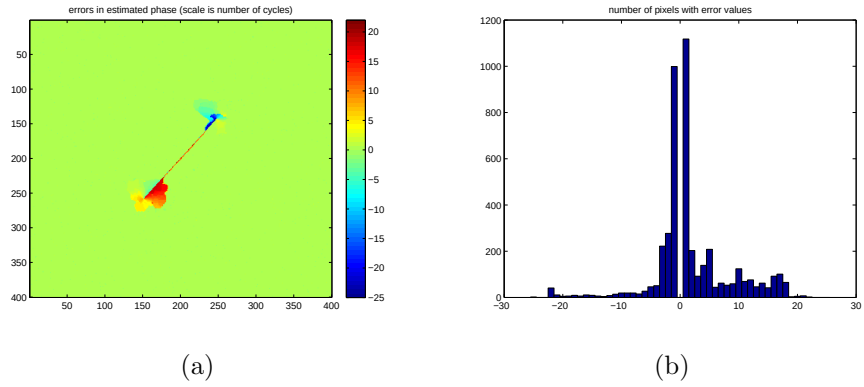


Figure 5.7: True error in the unwrapped solution. Since the data is simulated, it is possible to get a quantitative estimate of errors.

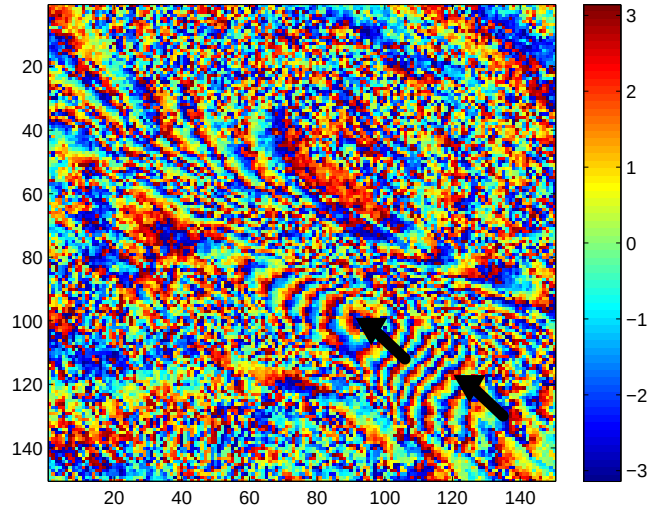


Figure 5.8: An example of real life interferogram. The arrows show locations of 2 pixels which are used for testing of unwrapped result. There are 8 cycles difference between these two locations.

The example chosen here is shown in Fig. 5.8.

This is then unwrapped with SNAPHU where cost functions are generated by SNAPHU itself. The output is shown in Fig. 5.9. The black lines shown in here are are branch-cuts made by SNAPHU.

In the resulted, unwrapped solution, the difference between pixels pointed by arrows is approximately 3 cycles which is clearly the direct consequence of branch cuts being put at wrong locations. The branch cuts are crossing the smooth fringe pattern which is undesirable since unwrapped solution will have phase jumps are these locations. Visually inspecting, there are no phase jumps at these locations.

Now, new cost functions are generated and unwrapping done again. The result of new unwrapped solution is shown in Fig. 5.10. Here, new branch cuts are not going across fringes so smooth part of the interferogram doesn't have any phase jumps which is desirable.

The difference between two pixels highlighted in Fig. 5.8 here is about 7 cycles

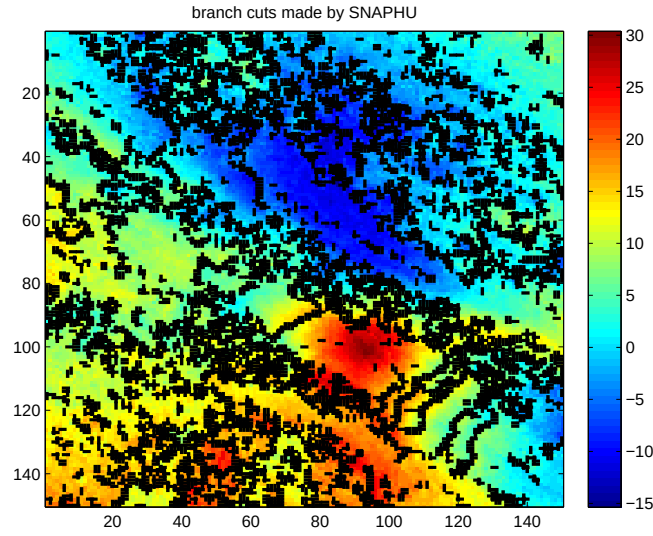


Figure 5.9: The unwrapped solution provided by SNAPHU. Branch cuts are also shown as black lines. As seen, SNAPHU is putting branch cuts across the smooth fringes.

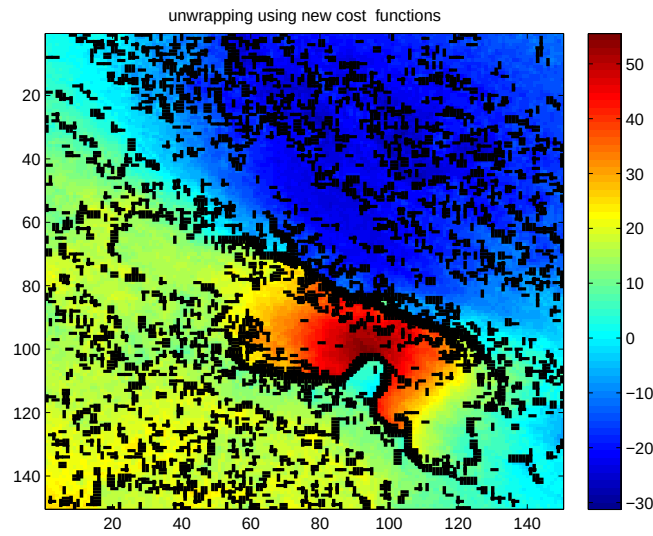


Figure 5.10: The unwrapped solution provided by SNAPHU after setting up new cost functions. Branch cuts are also shown. As seen, now the SNAPHU is putting branch cuts which are not crossing smooth fringes.

which is in better agreement with manual calculations. There are about 8 fringes between these two pixels so there should be about 8 cycles difference in unwrapped phase too. The unwrapping process depends greatly on input parameters which need to set empirically per interferogram. Edge detection in case of simulated data performs better if FCC is used for edge detection since the phase gradient is varying smoothly. On the other hand, real life data showed better results in detecting edges from phase gradients.

5.3 Limitations

It is not always possible to detect edges. It happens in cases where the fringe pattern is not changing abruptly and rather there is a smooth transition. Visual inspection of these types of simulation gives a better idea. Fig. 5.11 shows examples of cases where edge detection techniques didn't work.

These examples show that new method is promising. The shortcomings in terms of failure of edge detection can't be handled using Windowed Fourier Transformation method. Other techniques which might are not tested here and are left for future work.

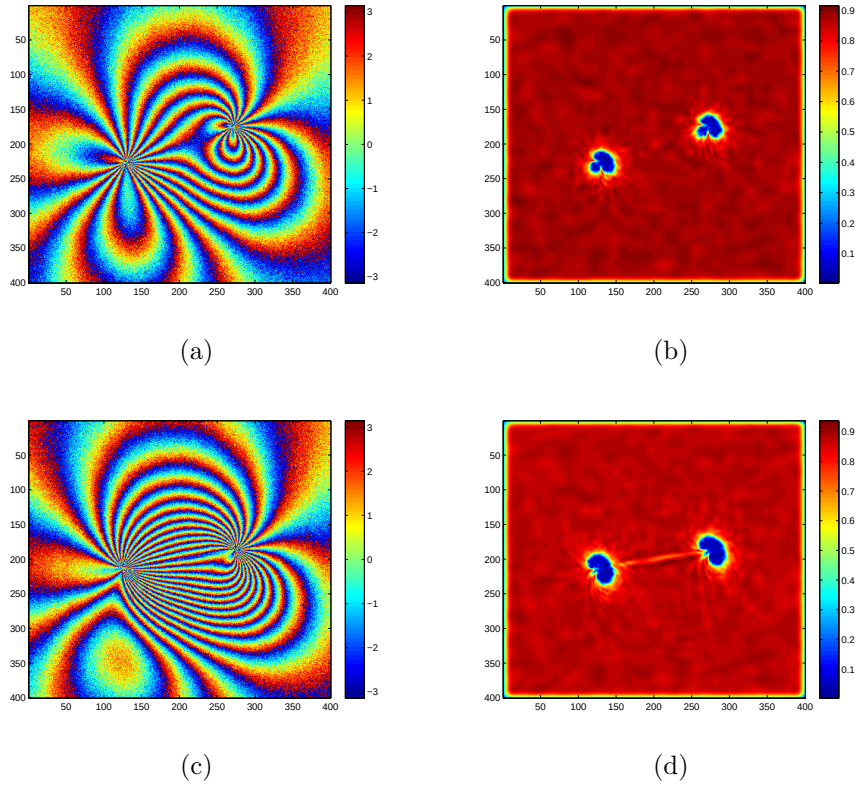


Figure 5.11: Example of cases where it is hard to detect edges since the fringe pattern remains smooth along the fault. Left images are simulated data and right images are their FCC. Since there is no clearly visible change in phase along the fault, the technique didn't work.

Chapter 6

Conclusion and Recommendations

The research objective at the beginning of the study was:

Improvement of statistical cost functions using additional parameters to improve phase unwrapping algorithms.

Based on the study conducted as a master's thesis, certain conclusions and recommendations are made which is given in following sections.

6.1 Conclusion

Under the light of the research objective, a new algorithm is proposed which generates improved cost functions for SNAPHU. This new method is based on estimation of phase gradients and discontinuities in phase gradients. It is observed that it is feasible in cases where the interferogram is less noisy and there are clearly visible phase discontinuities. On the other hand, there are cases in which the new method is not estimating phase discontinuities. These interferograms have smooth transition of wrapped phase over faults. Nevertheless, theoretical basis is stronger

than previously used in SNAPHU which was based only on coherence values. Detected edges sometimes have certain artifacts due to processing steps. Post processing is required to finalize the results from edge detection techniques. This post processing requires specific attention as per interferogram and the method is not generalized for all the cases. A general schematic is proposed which is common to all the cases though but input parameters needs to set up separately for each case.

The new algorithm is tested with both simulated and real data. Simulated data makes it possible to quantitatively assess the unwrapped solution. Once the results are acceptable, real data examples are tested. Since real data examples can't provide a mathematical estimate of quality, they are inspected visually and some manual calculations.

The goal was to develop a time efficient algorithm but under the given time constraints of thesis duration, algorithm still remain computationally expensive. The proposed algorithm iteratively estimates the phase gradients using different Fourier window functions. This procedure is computationally expensive and computation time increases with number of steps occurring during iteration. If the Fourier basis function is used according to the interferograms under consideration, the steps taken during iteration can be decreased and this increases efficiency of the algorithm. A generally low phase gradient interferogram doesn't require to be tested for high gradients. If the Fourier function is set to estimate lower gradients only, considerable amount of time can be saved.

The new proposed algorithm can be further improved by considering some more factors. A few recommendations are given in next section.

6.2 Recommendations

Several recommendations are made in this section:

1. WFT is not the only method that can give phase gradients and edges de-

tections. There are several other techniques which are based on Fuzzy logic (Tyan and Wang, 1993), Neural Networks (Waldemark et al., 2000), Wavelet transformation (Waldemark et al., 2000), etc. These techniques were not thoroughly inspected during current study. These techniques might provide a faster way for phase gradient analysis. It is recommended to check these options for further study.

2. Phase gradient estimation and edge detection step is slow. Nested iterations can be used to improve the speed. First, a coarse estimation can be made and then finer estimation at location where there is more likelihood of edge detection. At the same time, coarse estimation of phase gradients can be made.
3. MATLAB is used as a programming tool for the current study which is not a fast solver. Code efficiency will increase if algorithm is implemented in C or C++. An independent solver routine completely developed and optimized in C should be more computationally efficient.
4. SNAPHU works only on a regular grid. Recently, efforts have been made for unwrapping of sparse grids too (Agram and Zebker, 2009). A sparse grid is where the phase is not sampled at a regular interval in space. General principles of phase gradient estimation should be applicable in the case of sparse grid too. It is recommended to test proposed method for unwrapping on a sparse grid too.

Appendix A

SNAPHU configuration file

```
# snaphu configuration file
```

```
#####  
# File input and output and runtime options #  
#####
```

```
# See section below for file format configuration options.
```

```
# Input file name  
INFILE snaphu.in
```

```
# Output file name  
OUTFILE snaphu.out
```

```
# Input cost file (for statistical costs).  If costs are read from this  
# file, many of the other parameters will be ignored (string).  
# COSTINFILE snaphu.costinfile  
COSTOUTFILE snaphu.costoutfile
```

```
# Statistical-cost mode (TOP0, DEF0, SMOOTH, or NOSTATCOSTS)
```

```
STATCOSTMODE DEFO
#####
# Valid data formats:
#
# COMPLEX_DATA:      complex values: real, imag, real, imag
# ALT_LINE_DATA:     real values from different arrays, alternating by line
# ALT_SAMPLE_DATA:   real values from different arrays, alternating by sample
# FLOAT_DATA:        single array of floating-point data
#
INFILEFORMAT COMPLEX_DATA
#OUTFILEFORMAT ALT_LINE_DATA
OUTFILEFORMAT FLOAT_DATA
#corr file
CORRFILEFORMAT      FLOAT_DATA

#MAXFLOW             6
CS2SCALEFACTOR       15
# End of snaphu configuration file
```

This file contains the required input flags for the unwrapping operation.

Bibliography

- P.S. Agram and H.A. Zebker. Sparse two-dimensional phase unwrapping using regular-grid methods. *Geoscience and Remote Sensing Letters, IEEE*, 6(2):327–331, 2009.
- Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. ISBN 0-13-617549-X.
- D.P. Bertsekas. *Network optimization: continuous and discrete models*. Athena Scientific Belmont, Massachusetts, 1998.
- Y. Canny. A computational approach to edge detection. *IEEE transactions on pattern analysis and machine intelligence*, 8(6):679–698, 1986.
- C W Chen and H A Zebker. Phase unwrapping for large SAR interferograms: statistical segmentation and generalized network models. *IEEE Transactions on Geoscience and Remote Sensing*, 40(8):1709–1719, August 2002a.
- Curtis W Chen. *Statistical-cost network-flow approaches to two-dimensional phase unwrapping for radar interferometry*. PhD thesis, Stanford University, June 2001.
- Curtis W Chen and Howard A Zebker. Network approaches to two-dimensional phase unwrapping: intractability and two new algorithms. *Journal of the Optical Society of America A.*, 17(3):401–414, March 2000a. URL <http://nova.stanford.edu/~curtis/research/netunw.pdf>.
- Curtis W Chen and Howard A Zebker. Two-dimensional phase unwrapping using

- statistical models for cost functions in nonlinear optimization. *Journal of the Optical Society of America A.*, 18:338–351, February 2000b. URL <http://nova.stanford.edu/~curtis/research/statcost.pdf>.
- Curtis W Chen and Howard A Zebker. Phase unwrapping for large SAR interferograms: Statistical segmentation and generalized network models. *IEEE Transactions on Geoscience and Remote Sensing*, 40:1709–1719, August 2002b.
- N H Ching, D Rosenfeld, and M Braun. Two-dimensional phase unwrapping using a minimum spanning tree algorithm. *IEEE Transactions on Image Processing*, 1:355–365, 1992.
- Mario Costantini. A phase unwrapping method based on network programming. In ‘*FRINGE 96*’ workshop on ERS SAR Interferometry, Zürich, Switzerland, 30 Sep–2 October 1996, 1996.
- Mario Costantini. A novel phase unwrapping method based on network programming. *IEEE Transactions on Geoscience and Remote Sensing*, 36(3):813–821, May 1998.
- M. Dekking. *A Modern Introduction to Probability and Statistics: Understanding why and how*. Springer Verlag, 2005.
- Dennis C Ghiglia and Mark D Pritt. *Two-dimensional phase unwrapping: theory, algorithms, and software*. John Wiley & Sons, Inc, New York, 1998.
- Dennis C Ghiglia and Louis A Romero. Direct phase estimation from phase differences using fast elliptic partial differential equation solvers. *Optics Letters*, 14(20):1107–1109, October-15 1989.
- Dennis C Ghiglia and Louis A Romero. Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods. *Journal of the Optical Society of America A.*, 11(1):107–117, January 1994.
- Dennis C Ghiglia and Louis A Romero. Minimum l^p -norm two-dimensional phase unwrapping. *Journal of the Optical Society of America A.*, 13:1999–2013, 1996.
- A.V. Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of Algorithms*, 22(1):1–29, 1997.

- Richard M Goldstein, Howard A Zebker, and Charles L Werner. Satellite radar interferometry: Two-dimensional phase unwrapping. *Radio Science*, 23(4):713–720, July 1988.
- Ramon F Hanssen. *Radar Interferometry: Data Interpretation and Error Analysis*. Kluwer Academic Publishers, Dordrecht, 2001.
- B R Hunt. Matrix formulation of the reconstruction of phase values from phase differences. *Journal of the Optical Society of America A.*, 69(3):393–399, March 1979.
- Q Kemao. Two-dimensional windowed fourier transform for fringe pattern analysis: Principles, applications and implementations. *Optics and Lasers in Engineering*, 45(2):304–317, 2007. URL <http://linkinghub.elsevier.com/retrieve/pii/S0143816606000455>.
- S. Moon-Ho Song, S. Napel, N.J. Pelc, and G.H. Glover. Phase unwrapping of mr phase images using poisson equation. *Image Processing, IEEE Transactions on*, 4(5):667–676, 1995.
- Mark D Pritt. Phase unwrapping by means of multigrid techniques for interferometric SAR. *IEEE Transactions on Geoscience and Remote Sensing*, 34(3):728–738, May 1996.
- Kemao Qian, Hock Soon Seah, and Anand Asundi. Fault detection by interferometric fringe pattern analysis using windowed fourier transform. *Measurement Science and Technology*, 16(8):1582, 2005. URL <http://stacks.iop.org/0957-0233/16/i=8/a=006>.
- J. Serra. *Image analysis and mathematical morphology*. London.: Academic Press.[Review by Fensen, EB in: J. Microsc. 131 (1983) 258.] Review article General article, Technique Microscopy Staining, Mathematics, Cell size (PMBD, 185707888), 1982.
- SNAPHU configuration file. Snaphu configuration file. http://www-star.stanford.edu/sar_group/snaphu/snaphu.conf.full, 2001.
- J. Szumowski, W.R. Coshov, F. Li, and SF Quinn. Phase unwrapping in the

- three-point dixon method for fat suppression mr imaging. *Radiology*, 192(2): 555–561, 1994.
- H. Takajo and T. Takahashi. Least-squares phase estimation from the phase difference. *JOSA A*, 5(3):416–425, 1988.
- R.J. Trudeau. *Introduction to graph theory*. Dover Publications, 1993.
- C.Y. Tyan and P.P. Wang. Image processing-enhancement, filtering and edge detection using the fuzzy logic approach. In *Fuzzy Systems, 1993., Second IEEE International Conference on*, pages 600–605. IEEE, 1993.
- K. Waldemark, T. Lindblad, V. Bećanović, J.L.L. Guillen, and P.L. Klingner. Patterns from the sky: satellite image analysis using pulse coupled neural networks for pre-processing, segmentation and edge detection. *Pattern recognition letters*, 21(3):227–237, 2000.
- Howard A Zebker and Yanping Lu. Phase unwrapping algorithms for radar interferometry: residue-cut least-squares, and synthesis algorithms. *Journal of the Optical Society of America A.*, 15(3):586–598, March 1998.