SAR Phase Unwrapping Using Path-Based Least-Squares Phase Estimation and Region-Growing with Polynomial-Based Phase Prediction

Benjamin Brunson

A thesis submitted to the faculty of graduate studies in partial fulfillment of the requirements for the degree of Master of Applied Science

Graduate program in Earth and Space Science
York University
Toronto, Ontario

May 2019

Abstract

Differential SAR interferometry (DInSAR) has proven to be a processing approach that is well-suited to precisely identifying large-scale land deformation patterns. This is useful for many environmental monitoring applications, but the speckle noise and temporal decorrelation present in SAR images presents particular challenges in processing SAR images. This research focuses on the phase unwrapping problem, proposing two new approaches: Polynomial-Based Region-Growing Phase Unwrapping (PBRGPU), which expands upon the traditional region-growing approach to phase unwrapping; and Path-Based Least-Squares Phase Unwrapping (PBLSPU), which extends the least-squares phase unwrapping models in a path-based framework. Both algorithms were tested using simulated data and interferograms generated from RADARSAT-2 data. Both approaches significantly reduced the root mean square error compared to the algorithms they build from, and achieved a similar level of performance to the commonly-used SNAPHU algorithm without the need for masking low coherence areas.

Acknowledgements

I would like to express appreciation to York University, NSERC, the Canadian Space Agency, and PCI Geomatics for providing financial support for this research. In addition, I would like to thank the Canadian Space Agency for providing the SAR data used in this research and PCI Geomatics for the use of their software and technical support services.

I would like to thank my supervisor, Professor Jianguo Wang, and my supervisory committee member, Professor Baoxin Hu, for the support they have provided throughout this research and through all my studies. I would like to extend my thanks to Professor Mojgan Jadidi and Professor Juejiao Fu for having served on my M.A.Sc. defense committee. I would also like to thank my family for supporting and encouraging me through my studies.

Table of Contents

Abstract ii

Acknowledgements iii

Table of Contents iv

List of Tables v

List of Figures vi

1 Introduction 1

2 Literature Review 5

2.1 Introduction to the Phase Unwrapping Problem 7

2.2 General Approaches to Phase Unwrapping 8

2.3 Integration-Based Phase Unwrapping Methods 10

2.3.1 Branch-Cut Phase Unwrapping (BCPU) 13

2.3.2 Minimum Weighted Discontinuity Phase Unwrapping 16

2.3.3 Region-Growing Phase Unwrapping (RGPU) 17

2.3.4 SNAPHU 20

2.4 Least-Squares Phase Unwrapping 20

2.4.1 The Weighted Least-Squares Approach 24

3 Proposed Phase Unwrapping Methods 26

3.1 Polynomial-Based Region-Growing Phase Unwrapping (PBRGPU) 26

3.2 Path-Based Least-Squares Phase Unwrapping (PBLSPU) 32

4 Results 37

4.1 Simulated Data 37

4.1.1 Polynomial-Based Region-Growing 39

4.1.2 Path-Based Least-Squares Phase Unwrapping 41

4.2 RADARSAT-2 Data 42

4.2.1 Polynomial-Based Region-Growing Phase Unwrapping 45

4.2.2 Path-Based Least-Squares Phase Unwrapping 47

4.2.3 Comparison to SNAPHU Algorithm 50

4.2.4 Comparison Between Proposed Approaches 52

4.2.5 On Local Normality of the Interferometric Phase 53

5 Conclusion and Remarks 55

References 57

List of Tables

Table 4.1: Summary of the noise characteristics of the simulated signals used to test the phase unwrapping algorithms. The level of noise is characterized by the standard deviation of normally-distributed random noise that was added to the ideal simulated signal. 37

Table 4.2: Comparison of Region-Growing Phase Unwrapping (RGPU) and Polynomial-Based Region-Growing Phase Unwrapping (PBRGPU) results for the simulated data. 40

Table 4.3: Comparison of Path-Based Least-Squares Phase Unwrapping (PBLSPU) to standard Least-Squares Phase Unwrapping (LSPU) and Region-Growing Phase Unwrapping (RGPU) results for the simulated data. 41

Table 4.4: Comparison of PBLSPU and RGPU results for the interferograms derived from RADARSAT-2 data for Polar Bear Provincial Park. 45

Table 4.5: Comparison of PBLSPU, RGPU, and LSPU results for the interferograms derived from RADARSAT-2 data for Polar Bear Provincial Park 48

List of Figures

Figure 2.1: General InSAR/DInSAR workflow [Wessels 2017]……6
Figure 2.2: Example of a wrapped signal, with noise (colourmapped). Note the sharp discontinuities present in the signal……7
Figure 2.3: The unwrapped signal corresponding to the wrapped signal presented in Figure 2.2 (colourmapped)……7
Figure 2.4: A categorization of existing phase unwrapping methods……10
Figure 2.5: A wrapped one-dimensional signal, with positive discontinuities displayed as red points and negative discontinuities displayed as blue points……11
Figure 2.6: The results of unwrapping the noisy wrapped signal in Figure 1 using different pre-defined paths of integration. The image on the left shows the result of using range-wise (row-wise) integration, and the image on the right shows the result of using azimuth-wise (column-wise) integration. Note the horizontal streaking for range-wise integration and the vertical streaking for azimuth-wise integration……12
Figure 2.7: Examples of a positive residue (left), a pixel that is not a residue (middle), and a negative residue (right). Positive discontinuities are shown with red vectors, negative discontinuities are shown with blue vectors, and connections with no discontinuity are shown with green vectors. The upper-left pixel corresponds to ϕ_{ij} in (2.3.2)……14
Figure 2.8: An example showing the net discontinuity about a loop containing both a positive residue (red point) and negative residue (blue point)……14
Figure 2.9: A portion of the wrapped signal, with positive residues shown as red points and negative residues shown as blue points……14
Figure 2.10: Multiple configurations of residue cuts corresponding to the same residue locations/values [Chen and Zebker 2000]. The residue cuts shown in the left image correspond to L-0 norm minimization and the right image shows the results of forming residue cuts using Huntley's [1989] method……15
Figure 2.11: The results of applying Huntley's BCPU algorithm to the wrapped signal in Figure 1 (left) and the results of applying the algorithm after additional noise has been injected into the wrapped signal (right). Note the streaked patterns in the right image……15
Figure 2.12: Comparison of unwrapping results using the same set of branch cuts but different starting points for the integration. The unwrapped signal on the left used a starting point in the upper-left corner of the interferogram and the unwrapped signal on the right used a starting point in the centre of the interferogram.
Figure 2.13: Example of the flood-fill based seeded region growing used in the RGPU method. Seeded regions are represented with green cells, and yellow cells represent pixels being unwrapped……17
Figure 2.14: Depiction of the prediction lines around a pixel being unwrapped. The yellow cell represents the pixel being unwrapped, the green cells represent pixels in the neighbourhood that can factor into phase prediction, and the red cells represent pixels that cannot factor into phase prediction……17
Figure 2.15: Errors in the LSPU solution for the wrapped signal presented in Figure 1……23
Figure 2.16: A portion of the LSPU errors shown in Figure 13, with residue and residue cut information superimposed. Positive and negative residues are red and blue circles, respectively, and residue cuts are black lines connecting positive and negative residues……23
Figure 3.1: Visualization of the polynomial-based phase estimation process for the pixel being unwrapped (marked by the red dot). The unwrapped phase values in the neighbourhood of the pixel are retrieved (left), then a bivariate polynomial model is fit to the unwrapped phases

(middle), and this model is used to predict the unwrapped phase for the pixel being unwrapped (right). *28*
Figure 3.2: Local variance in a simulated wrapped interferogram. Notice the large spikes in variance near discontinuities in the interferogram. *29*
Figure 3.3: Local variance in a simulated wrapped interferogram after thresholding is applied. The thresholded regions are not considered in the estimation of the a priori variance of the wrapped phases. *29*
Figure 3.4: A Priori variance of a simulated wrapped interferogram. *30*
Figure 3.5: Flow chart of phase estimation in the Polynomial-Based Region-Growing Phase Unwrapping (PBRGPU) algorithm. *31*
Figure 3.6: Example of the LSPU matrix equations being applied directly for an 11 x 11 patch in a wrapped interferogram. It is important to note that the horizontal and vertical phase differences constitute different observations when applying the matrix equations directly, rather than them combining to form a single quantity as is the case when using DCTs to resolve the least-squares unwrapped phases. *33*
Figure 3.7: Illustration of why the "connectedness" of pixels in the solution is important. The top image shows inlying/outlying pixels, and outlying pixels result in there being a disconnected region in the upper-left corner of the interferogram patch. The bottom image shows the pixels that are included in the least-squares solution. Note that the region in the upper left corner of the patch is excluded. *34*
Figure 3.8: Flowchart of phase estimation in the path-based least-squares phase unwrapping algorithm *36*
Figure 4.1: The simulated wrapped signals used to test the proposed phase unwrapping algorithms. The top-left signal has no noise, the top-right has low noise, the bottom left has medium noise, and the bottom right has high noise..... *38*
Figure 4.2: The ideal unwrapped signals corresponding to the wrapped signals in Figure 4.1. The top-left signal has no noise, the top-right has low noise, the bottom left has medium noise, and the bottom right has high noise..... *39*
Figure 4.3: The differences between the PBRGPU unwrapped signal and the ideal unwrapped signal (left) and the differences between the RGPU unwrapped signal and the ideal unwrapped signal (right) for the high noise level simulated interferogram..... *41*
Figure 4.4: The differences between the PBLSPU unwrapped signal and the ideal unwrapped signal (left)and the differences between the LSPU unwrapped signal and the ideal unwrapped signal (right) for the high noise level simulated interferogram..... *42*
Figure 4.5: High coherence interferogram. The coherence values for the interferogram are shown in the left image, and the wrapped phase values are shown in the right image. Images were collected on June 23 and July 17, 2011..... *43*
Figure 4.6: Low coherence interferogram. The coherence values for the interferogram are shown in the left image, and the wrapped phase values are shown in the right image. Images were collected on June 23 and September 3, 2011..... *44*
Figure 4.7: Varied coherence interferogram. The coherence values for the interferogram are shown in the left image, and the wrapped phase values are shown in the right image. Images were collected on June 23 and August 10, 2011..... *44*
Figure 4.8: Unwrapped phase determined for the high coherence interferogram using PBRGPU (left) and RGPU (right), in radians. Excluded pixels are shown in white..... *46*
Figure 4.9: Unwrapped phase determined for the low coherence interferogram using PBRGPU (left) and RGPU (right), in radians. Excluded pixels are shown in white..... *46*

Figure 4.10: Unwrapped phase determined for the varied coherence interferogram using PBRGPU (left) and RGPU (right), in radians. Excluded pixels are shown in white……..47
Figure 4.11: Unwrapped phase determined for high coherence interferogram using the PBLSPU (left) and LSPU (right) methods, in radians. Excluded pixels are shown in white……48
Figure 4.12: Unwrapped phase determined for low coherence interferogram using the PBLSPU (left) and LSPU (right) methods, in radians. Excluded pixels are shown in white……49
Figure 4.13: Unwrapped phase determined for varied coherence interferogram using the PBLSPU (left) and LSPU (right) methods, in radians. Excluded pixels are shown in white……..49
Figure 4.14: SNAPHU unwrapped phase values for the high coherence interferogram, in radians……..51
Figure 4.15: SNAPHU unwrapped phase values for the varied coherence interferogram, in radians……51
Figure 4.16: SNAPHU unwrapped phase values for the low coherence interferogram, in radians……52
Figure 4.17: Comparison between the histogram of differences between predicted and unwrapped phase values for the high coherency interferogram and a T-distribution with 3 degrees of freedom for the PBRGPU algorithm……53
Figure 4.18: The degrees of freedom of the polynomial fitting process for interferogram patches in the high coherency interferogram……54

1 Introduction

Synthetic Aperture RADAR (SAR) is an active microwave remote sensing technology that gathers information about the scattering properties of Earth's surface [Pepe and Calò 2017]. The scattering properties described by SAR images from satellite-mounted SAR systems (such as RADARSAT, the ERS-1/2 satellites, and the Sentinel-1 constellation) are particularly useful for environmental applications that are linked to moisture content, including the mapping of soil moisture content [Alexakis et al 2016], marshlands [Cazals et al 2016], water bodies [Li and Wang 2017], and glaciers and their subsurface properties [Winsvold et al 2017]. These environmental monitoring applications may be addressed through the use of single-pass SAR images, but integrating data from multiple satellite passes allows for monitoring of how these quantities change through time. Moreover, the phase differences between multiple-pass SAR images are related to the topography of the study area, and may be used to generate digital elevation models through Interferometric SAR (InSAR) analysis, as well as deformation maps through Differential SAR Interferometry (DInSAR).

Precise, large-scale land deformation analysis has important environmental applications, including monitoring permafrost level changes in Arctic regions and assessing land subsidence around regions of dense urban construction. Such large-scale analysis is challenging and time-consuming to address using traditional land surveying techniques, especially for large areas of study. Most satellite imagery lacks the accuracy required for this precision analysis, but applying DInSAR processing techniques to SAR images makes this analysis more feasible. DInSAR provides precise deformation products that have large areas of coverage without requiring extensive fieldwork. The main objective of this research is to improve DInSAR processing techniques with a particular focus on addressing the issue of phase unwrapping.

The DInSAR process derives deformation information from pixel-wise phase difference maps (interferograms) between SAR images collected over the course of several months [Pepe and Calò 2017]. Before producing usable deformation products, there are several pre-processing tasks that must be performed. The DInSAR process consists of the following main procedures:

1) SAR Image Alignment:
    - An important part of integrating information from multiple SAR images is ensuring that each pixel of those images refers to the same geographic area. This requires the images to be aligned to a high degree of precision, usually to the sub-pixel level. The process of SAR

image alignment is typically divided into a coarse alignment which aligns images to the pixel level (typically using keypoint matching), followed by a fine alignment which aligns images to the sub-pixel level (typically using oversampling and Fourier transformations to achieve 1/20 pixel precision) [Pepe and Calò 2017].

2) Interferogram Generation, Correction, and Filtering:
   - Once a pair of SAR images is aligned, the pixel-wise phase differences between the SAR images may be calculated, forming the interferogram of the SAR image pair. The interferogram may be used to derive deformation products, but local topography and orbital errors have systematic effects on the interferogram that must be accounted for. SAR interferograms are typically very noisy, and it is common practice to apply Goldstein filtering to the interferogram before performing phase unwrapping [Pepe and Calò 2017].
3) Phase Unwrapping:
   - The calculated phase differences in the interferogram are restricted to have values between $-\pi$ and π radians, which produces discontinuous regions in the interferogram. This phenomenon is known as "phase wrapping," and many algorithms have been proposed to "unwrap" interferograms and resolve these discontinuities [Pepe and Calò 2017].
4) Deformation Product Creation:
   - After an interferogram has been corrected, filtered, and unwrapped, geometric relationships may be used to derive deformation products for the study area [Pepe and Calò 2017].

The issue of phase unwrapping presents a particular challenge in the DInSAR process due to the presence of "speckle noise" in SAR interferograms. SAR sensors measure the backscattered microwave pulses, and speckle noise is produced by the backscattered signals interfering with one another [Pepe and Calò]. Speckle noise consists of an additive component and a multiplicative component, and has a significant detrimental impact on automated phase unwrapping algorithms. For a phase unwrapping algorithm to be able to produce accurate results, it must account for the high noise levels in SAR interferograms. Also, temporal decorrelation between SAR images can occur when there are significant changes to surface objects (e.g. vegetation growth, construction, etc.). This also presents a challenge in the context of phase unwrapping when using multi-pass SAR images, since each SAR image is recorded over a different topographic surface. SAR coherence maps provide a means of describing interferogram quality, with a low coherence indicating high noise levels or temporal decorrelation.

Existing phase unwrapping algorithms either use path-based integration or global optimization to resolve discontinuities in an interferogram. Both types of approaches have inherent limitations that make them quite challenging to apply to noisy SAR interferograms.

Path-based approaches to phase unwrapping tend to be highly susceptible to the speckle noise present in SAR interferograms, and individual unwrapping errors tend to propagate along the integration path as the algorithm progresses. Even very successful path-based approaches, such as the popular SNAPHU approach, struggle in regions that have particularly high noise levels. Some phase unwrapping approaches attempt to mitigate this issue by assessing the quality of the unwrapped signal, but the quality assurance measures these approaches use lack the robustness required for high noise SAR interferograms.

Global optimization approaches to phase unwrapping typically use L-2 norm minimization, which is a technique that can amplify the effects outlying pixels have on the unwrapping solution, particularly when there is a high proportion of outlying pixels, as is the case for SAR images. In addition, current realizations of global optimization approaches rely on using discrete cosine transforms, which does not allow for the detection and removal of high-noise pixels. The combination of these two issues makes global optimization approaches produce large unwrapping errors when applied to SAR interferograms.

The primary objectives of this research are:

1) To build upon and modify existing phase unwrapping approaches to increase the overall unwrapping accuracy;
2) To build quality assurance and removal of high-noise pixels into a global optimization phase unwrapping approach, overcoming one of the main limitations of traditional global optimization approaches;
3) To use rigorous statistical testing to perform ongoing quality assurance as an interferogram is being unwrapped, improving upon currently-defined quality assurance measures; and
4) To structure the proposed approaches around the high noise levels present in SAR interferograms and to produce accurate unwrapped signals in high noise environments.

Through this research, two novel phase unwrapping approaches are proposed, implemented and analysed: Polynomial-Based Region-Growing Phase Unwrapping (PBRGPU) and Path-Based Least-Squares Phase Unwrapping (PBLSPU). The former is structured after the Region-Growing Phase Unwrapping (RGPU) approach, but uses polynomial fitting for improved phase prediction and rigorous quality assurance. The latter builds upon the matrix formulation of Least-Squares Phase Unwrapping

(LSPU) to provide statistically rigorous quality assurance and to allow for the complete removal of detected high-noise pixels, which are not possible to remove using the traditional LSPU approach.

Both approaches make significant improvements to the robustness of existing phase unwrapping algorithms. In particular, the PBLSPU approach allows for high-noise pixels to be completely removed from the unwrapping solution, which was previously not possible when using a global optimization approach to phase unwrapping. Both approaches make extensive use of statistical quality assurance, which has two main advantages. Firstly, the statistical quality assurance tests are adaptive to local noise levels in the interferogram. Secondly, they allow for rejection criteria to be defined in terms of statistical significance levels, which are much more intuitive to define than separate threshold parameters.

While the proposed algorithms were developed for SAR interferograms, they could be applied to unwrap any wrapped signal. Phase unwrapping is necessary for field mapping in Magnetic Resonance Imaging (MRI) [Ghiglia and Romero 1996], high-accuracy x-ray profiling [Ying 2006], and the measurement of wavefront distortions in adaptive optics [Ying 2006].

This introduction is followed by a literature review (Chapter 2) with a description of the phase unwrapping problem. It also includes a discussion of each algorithm's strengths, as well as some of the challenges that still need to be overcome.

Further, Chapter 3 proposes the following two phase unwrapping algorithms: Polynomial-Based Region-Growing Phase Unwrapping (PBRGPU) and Path-Based Least-Squares Phase Unwrapping (PBLSPU).

Chapter 4 compares the performance of the proposed algorithms with those commonly used in practice. This is accomplished using both simulated wrapped signals (which allow for the assessment of the root mean square errors in the unwrapped signals) and interferograms that were generated using RADARSAT-2 data. This chapter also discusses the validity of some of the assumptions that were made in the development of the proposed algorithms, and how this may affect the quality of the unwrapping solutions they produce.

Chapter 5 summarizes the key findings of this research, and proposes possible avenues for improvement to the proposed approaches.

2 Literature Review

The InSAR and DInSAR processes use the pixel-wise phase differences between complex-valued SAR images (interferograms) to produce topographic/deformation products. There are several factors affecting these phase differences that must be accounted for when producing topographic/deformation products from an interferogram: differences in geometries between SAR images being processed (image misalignment); topographic effects (when producing deformation products); effects of errors in the satellites' orbital parameters; and discontinuities resulting from the phenomenon of phase wrapping [Pepe and Calò 2017].

The issue of SAR image misalignment requires the precise alignment of any SAR image pair being processed, typically to the sub-pixel level [August et al 2010, Li and Bethel 2008, Sun and Muller 2016]. Once image pairs have been precisely aligned, a sufficiently accurate digital elevation model of the study area may be used to resolve topographic effects on the calculated phase differences [Pepe and Calò 2017]. The errors in satellites' orbital parameters often manifest themselves as linear fringe patterns in the phase differences [Bähr and Hanssen 2012], and these patterns may be removed in the spectral domain [Čapková 2005, Tian et al 2018]. Refer to Figure 2.1 for a complete workflow of the InSAR/DInSAR process.

Once these systematic effects are removed from a SAR interferogram, the discontinuities resulting from the phase wrapping phenomenon must be resolved via the process of phase unwrapping. While phase wrapping is simple to resolve in a one-dimensional signal, the process becomes significantly more complicated as the dimensionality of the problem increases [Gonzalez and Jacques 2014]. The problem is further complicated by the presence of discontinuous topographical features, SAR image noise, and temporal decorrelation between SAR images [Gonzalez and Jacques 2014]. This requires phase unwrapping approaches to be very robust to high noise levels, which is the main challenge to overcome when developing a viable phase unwrapping solution.

SAR images are complex-valued, and complex-valued interferograms are produced by taking the product of one SAR image with the complex conjugate of the other. The phase of the complex interferogram describes the wrapped phase values, and the amplitude of the complex interferogram describes the coherence, which describes the degree of agreement between the SAR images.

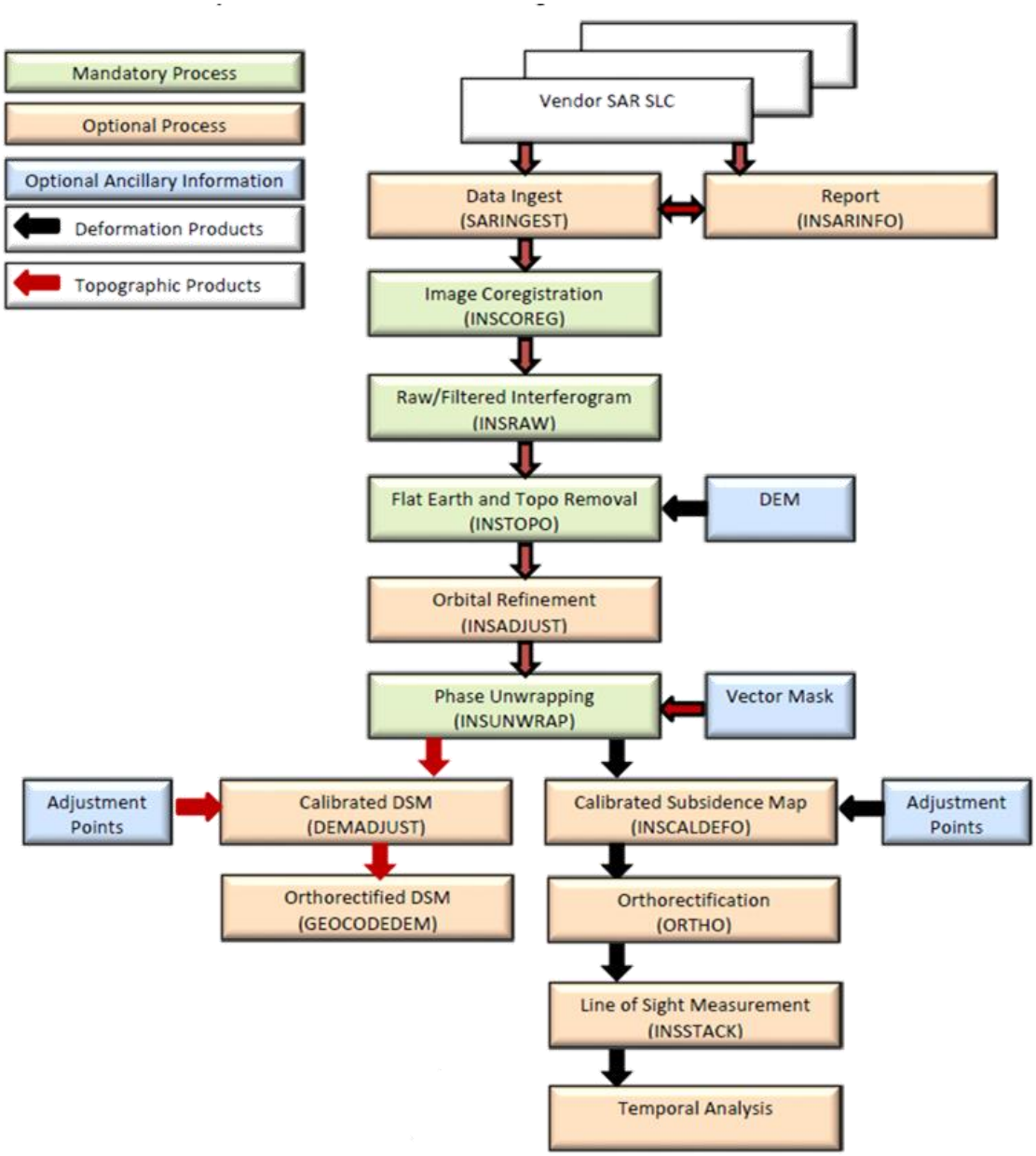

Figure 2.1: General InSAR/DInSAR workflow [Wessels 2017]

2.1 Introduction to the Phase Unwrapping Problem

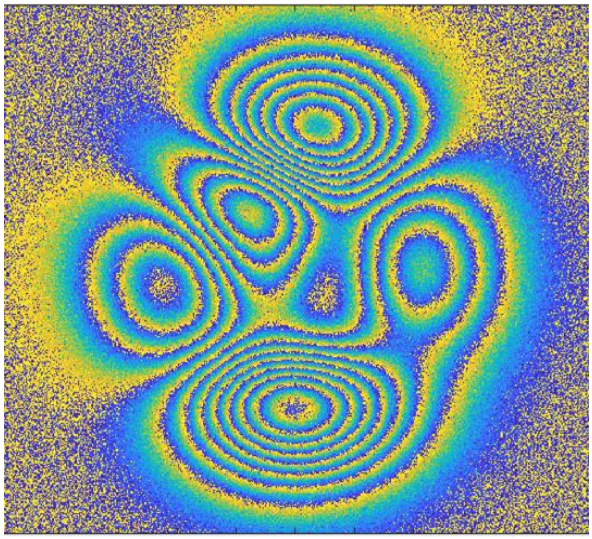

Figure 2.2: Example of a wrapped signal, with noise (colourmapped). Note the sharp discontinuities present in the signal.

Each phase difference value in the "true", or "unwrapped", interferogram may be considered to have two components: an integer number of wavelengths, and a fractional wavelength component. The calculation of the interferometric phase only provides information about the fractional component, and the integer number of wavelengths is ambiguous for every pixel in the interferogram [Pepe and Calò 2017]. This ambiguity is similar to the carrier phase ambiguity that arises in GPS processing, but there is a separate ambiguity parameter defined for each pixel in the interferogram. The ambiguities in the calculated phase values produces sharp, discontinuous boundaries in the interferometric phase, such as those seen in Figure 2.2. This phenomenon is known as "phase wrapping", and the interferometric phase must be "unwrapped" to derive useful information from it. The unwrapping process resolves discontinuities, producing a smoother signal such as the signal seen in Figure 2.3.

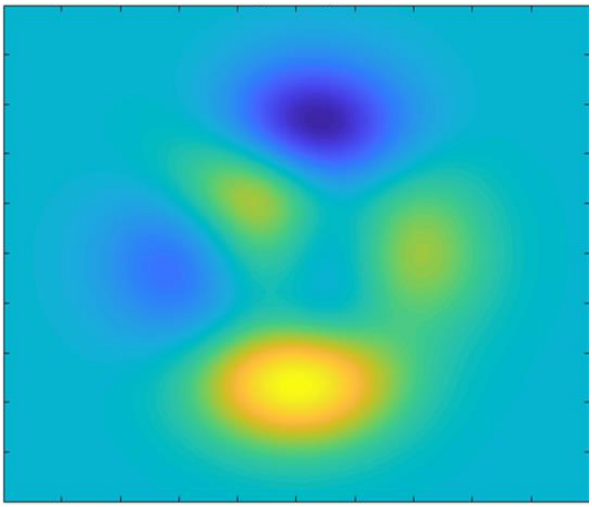

Figure 2.3: The unwrapped signal corresponding to the wrapped signal presented in Figure 2.2 (colourmapped).

The relationship between the wrapped and unwrapped phase is governed by the wrapping function, defined as

$$\phi_{ij} = \{\psi_{ij} - 2\pi n \mid n \in \mathbb{Z} \wedge -\pi < \phi_{ij} < \pi\} \tag{2.1.1}$$

where ϕ_{ij} and ψ_{ij} denote the wrapped and unwrapped phases, respectively, and n describes the integer ambiguity parameter associated with pixel location (i, j).

The large amount of speckle noise present in SAR images complicates the phase unwrapping process. One common approach applied to mitigate the effects of speckle noise is to smooth the interferogram to smooth it prior to the unwrapping process [Ferretti et al, 2007]. This approach reduces

noise levels in the interferometric phase, making the signal much easier to unwrap, but this reduced noise comes at the cost of spatial resolution in the unwrapped interferogram. While this trade-off may be acceptable for many large-scale topography/deformation studies, it is ill-suited to the spatial accuracy requirements for many engineering projects.

Multi-look SAR interferometry can further complicate the phase unwrapping process due to the potentially large time difference between SAR image recordings (time to revisit an area is typically about one month for SAR satellites). Over this period of time, land cover in the study area may change significantly (accumulating snow, leaf growth, etc.). This causes temporal decorrelation between SAR images, and the reduced correspondence between SAR images produces noisier interferometric phase values.

In light of these issues, a phase unwrapping algorithm must be robust to high noise levels in order to be successful. This is a challenging goal to achieve; integration-based algorithms are highly sensitive to outlying pixels, while global optimization algorithms spread the effects of outlying pixels on their surrounding regions.

2.2 General Approaches to Phase Unwrapping

Most phase unwrapping algorithms impose logical constraints on the relationships between adjacent unwrapped pixels in the interferogram in order to resolve the integer ambiguity parameter in (2.1.1). One commonly used approach is to impose a smoothness constraint on the unwrapped interferogram (i.e. assuming the topographic surface or deformation map to be continuous – in situations where this is not the case, discontinuity lines may be defined and used in the same way as in generating contours lines for topographic mapping). This constraint is imposed upon the wrapped differences between adjacent phase values in the interferogram. Here, the differencing operator is denoted $\Delta^{(a)}$ and $\Delta^{(r)}$ for azimuth and range, respectively, and their wrapped counterparts are indicated using a subscript w. The wrapped differencing operators are then defined as

$$\Delta_w^{(a)}\phi_{ij} = \begin{cases} \Delta^{(a)}\phi_{ij} & | -\pi \le \Delta^{(a)}\phi_{ij} \le \pi \\ \Delta^{(a)}\phi_{ij} + 2\pi & | \Delta^{(a)}\phi_{ij} < -\pi \\ \Delta^{(a)}\phi_{ij} - 2\pi & | \Delta^{(a)}\phi_{ij} > \pi \end{cases} \tag{2.2.1}$$

for wrapped phase differences in the azimuth and

$$\Delta_w^{(r)}\phi_{ij} = \begin{cases} \Delta^{(r)}\phi_{ij} & | -\pi \le \Delta^{(r)}\phi_{ij} \le \pi \\ \Delta^{(r)}\phi_{ij} + 2\pi & | \Delta^{(r)}\phi_{ij} < -\pi \\ \Delta^{(r)}\phi_{ij} - 2\pi & | \Delta^{(r)}\phi_{ij} > \pi \end{cases} \tag{2.2.2}$$

for wrapped phase differences in the range.

Ideally, the wrapped phase differences between pixels in the interferogram are equal to the phase differences between the pixels once they have been unwrapped. This leads quite naturally to the L-P norm cost function for the phase unwrapping process

$$C = \sum_{i,j} w_{ij}^{(a)} \left| \Delta^{(a)} \psi_{ij} - \Delta_w^{(a)} \phi_{ij} \right|^p + \sum_{i,j} w_{ij}^{(r)} \left| \Delta^{(r)} \psi_{ij} - \Delta_w^{(r)} \phi_{ij} \right|^p \quad (2.2.3)$$

where w_{ij} denotes the relative weights between the wrapped phase differences.

When p is 2, (2.2.3) becomes the cost function for a standard least-squares adjustment model. This forms the basis for the Least-Squares Phase Unwrapping (LSPU) method.

When p is 1, (2.2.3) describes the cost function used for many integration-based phase unwrapping algorithms, including the Branch-Cut Phase Unwrapping (BCPU) and Minimum Weighted Discontinuity (MWD) methods. It is worth noting that despite these methods using an L-1 norm cost function, they do not actually constitute L-1 norm minimizing algorithms; rather, they use the L-1 cost function as a means of detecting discontinuities between neighbouring pixels/discontinuous regions in the interferogram because it is well known that L-1 norm minimizing algorithms are superior to the L-2 norm minimizing algorithms for detecting high-noise measurements [Bektas and Sisman, 2010].

The special case that arises when p is 0 corresponds to an L-0 norm minimization method. When p is 0, the summation terms in (2.2.3) are only non-zero when there is an identified discontinuity in the interferogram. An L-0 norm minimization method would therefore minimize the total number of discontinuities in the unwrapped interferogram [Gao et al 2017]. As a result, many consider this to be the ideal optimization strategy for phase unwrapping [Chen and Zebker 2000]. Unfortunately, L-0 norm minimization is an NP-hard problem [Chen and Zebker 2000], which makes it intractable in practice. There are several methods that seek to approximate the L-0 norm minimization strategy, such as the Minimum Spanning Tree (MST) method employed by the SNAPHU algorithm [Chen and Zebker 2000].

Phase unwrapping approaches can be separated into two broad categories: path-based integration methods and global optimization methods (see Figure 2.4 for a breakdown of phase unwrapping methods). The main global optimization method in common use is the LSPU algorithm. Additionally, there are several methods that apply optimization strategies non-globally (such as the Kalman Filtering Phase Unwrapping algorithm [Xie 2016]), but these methods adhere to the general strategy of the path-based integration phase unwrapping methods.

Path-based integration methods share a similar processing workflow. The process begins by visiting a wrapped pixel, then using unwrapped pixels(s) in its neighbourhood to predict its unwrapped

phase. The difference between the wrapped phase and its predicted value is used to unwrap the pixel being visited. This pixel-by-pixel process is repeated for each wrapped pixel in the interferogram until all pixels along the path of integration have been unwrapped or visited by the algorithm.

Path-based integration methods can be categorized by when the integration path is defined or by how phase prediction is performed. The integration path can be determined prior to unwrapping the phase or it can be determined on-the-fly as the unwrapping algorithm progresses, and phase prediction can be done using a single unwrapped neighbouring pixel or using a function of all unwrapped pixels in the neighbourhood around the pixel being unwrapped.

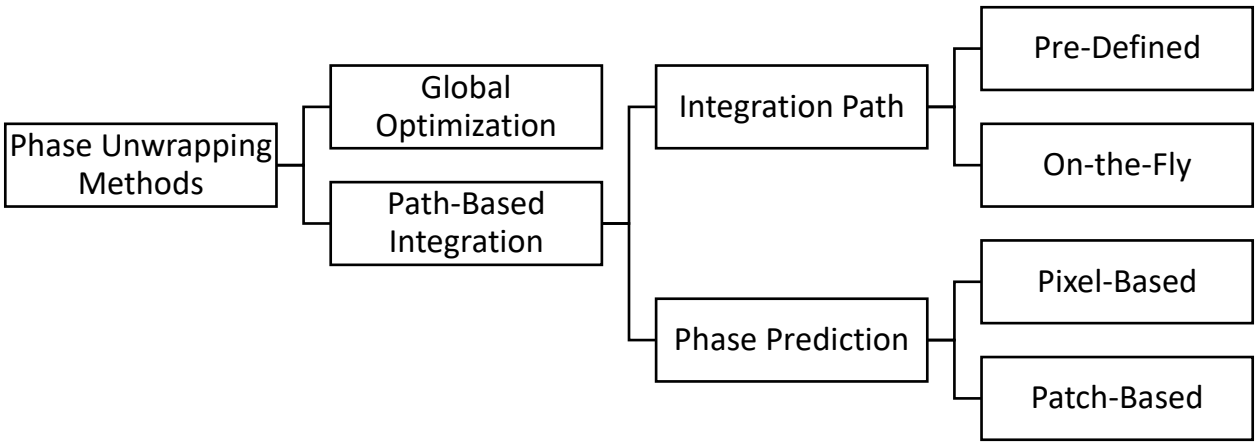

Figure 2.4: A categorization of existing phase unwrapping methods.

2.3 Integration-Based Phase Unwrapping Methods

Integration-based phase unwrapping methods rely on identifying and resolving discontinuities along a path of integration that spans the whole interferogram. This approach is quite successful for one-dimensional phase unwrapping, and the two-dimensional methods are formed by expanding the one-dimensional method. As such, the discussion of two-dimensional integration-based phase unwrapping will be prefaced with a discussion of its one-dimensional analogue.

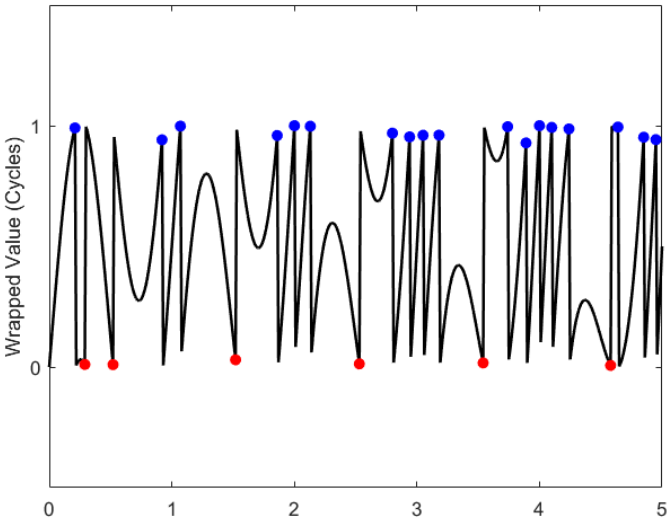

Figure 2.5: A wrapped one-dimensional signal, with positive discontinuities displayed as red points and negative discontinuities displayed as blue points.

For a one-dimensional signal, the difference between adjacent wrapped phases may be used to determine locations of discontinuities. By definition, a positive discontinuity occurs when a phase value is greater than half a cycle *more* than the preceding phase value. Similarly, a negative discontinuity occurs when a phase value is greater than half a cycle *less* than the preceding phase value. Figure 2.5 shows positive and negative discontinuities detected along the integration path of a simulated one-dimensional wrapped signal.

Discontinuities along the integration path can be resolved by determining the integer number of cycles that separates the phases on either side of the discontinuity. This integer may be calculated using the equation

$$m = round\left(\frac{\psi_{i-1} - \phi_i}{2\pi}\right) \tag{2.3.1}$$

where m denotes the integer number of cycles separating the unwrapped phase ψ_{i-1} from the wrapped phase ϕ_i [Xu and Cumming, 1999]. ϕ_i may now be unwrapped by adding m cycles to it.

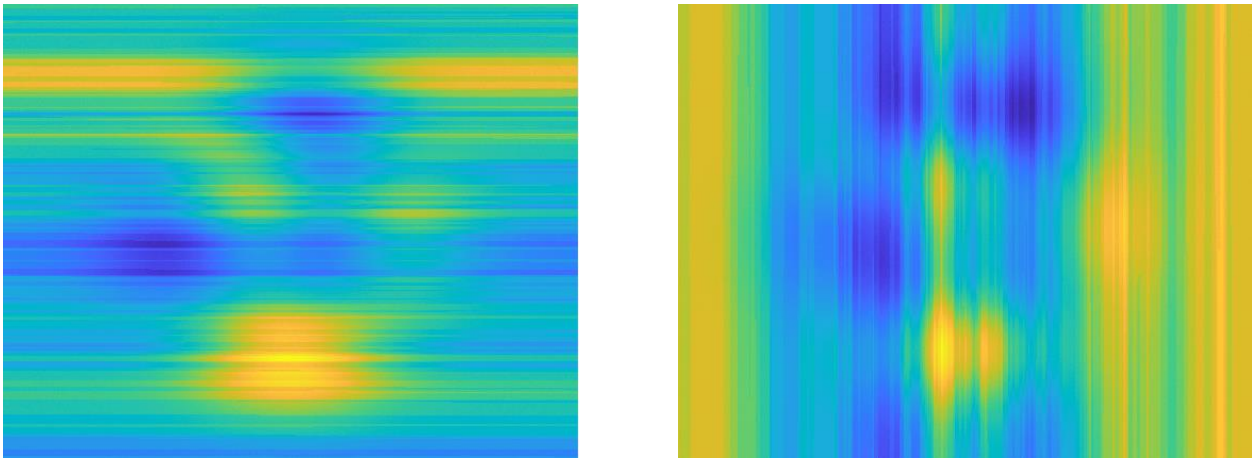

Figure 2.6: The results of unwrapping the noisy wrapped signal in Figure 1 using different pre-defined paths of integration. The image on the left shows the result of using range-wise (row-wise) integration, and the image on the right shows the result of using azimuth-wise (column-wise) integration. Note the horizontal streaking for range-wise integration and the vertical streaking for azimuth-wise integration.

The concept of detecting and resolving discontinuities using (2.3.1) remains consistent in two-dimensional integration-based phase unwrapping methods. In practice, this approach is complicated by the fact that many integration paths may be taken over two-dimensional signals and that values for the unwrapped phases are dependent upon the integration path taken. The unwrapping results are path-dependent due to false positive identification of discontinuities in noisy areas (the dependency of the unwrapping results on the integration path is illustrated in Figure 2.6). These falsely identified discontinuities are then "corrected" in the unwrapping process, causing unwrapping errors to propagate along the path of integration. Since unwrapping results are path-dependent, many path-based integration approaches to phase unwrapping are also affected by the starting point(s) of the integration.

The path-dependency of integration-based phase unwrapping makes it challenging to identify a unique solution for the unwrapped signal. To overcome the issue of non-uniqueness the unwrapping solution must be formed such that, given the unwrapped signal for one pixel, the unwrapped signal is uniquely defined everywhere else in the interferogram. There are two common ways of approaching this constraint:

a) By restricting the path of integration using pre-processing strategies. This is the strategy used in residue-cut algorithms, such as the Branch-Cut Phase Unwrapping (BCPU) algorithm.

b) By removing pixels that create inconsistencies from the integration path on-the-fly. This is the strategy used in the Region-Growing Phase Unwrapping (RGPU) method.

Approach (a) is the more theoretically rigorous approach which, if executed carefully, will satisfy the uniqueness constraint. Approach (b) lends itself more readily to techniques that adapt to local noise

levels in the interferogram, but its adherence to the uniqueness constraint heavily depends upon the selection of starting points for the integration as well as the methods used to identify inconsistencies.

2.3.1 Branch-Cut Phase Unwrapping (BCPU)

Huntley [1989] developed the Branch-Cut Phase Unwrapping (BCPU) algorithm to address the issues resulting from the path-dependency of unwrapping results. This algorithm effectively forms "cut lines" that the path of integration is not permitted to cross, and then uses flood fill-based integration to define paths of integration that flow around these cut lines. This method uses a pre-defined integration path with pixel-based phase prediction.

The BCPU method assumes the phase field to be conservative, and this leads directly to the condition that the line integral about any closed loop should be equal to zero. The interferogram's adherence to this condition may be tested by determining the net discontinuity about 2x2 loops for each pixel in the interferogram. For pixel (i,j), the condition is tested using pixels (i,j), $(i,j+1)$, $(i+1,j+1)$, and $(i+1,j)$. The net discontinuity about this loop is denoted s_{ij} and is given by the equation [Huntley 1989]

$$s_{ij} = round\left(\frac{\phi_{i+1,j} - \phi_{ij}}{2\pi}\right) + round\left(\frac{\phi_{i+1,j+1} - \phi_{i+1,j}}{2\pi}\right) + round\left(\frac{\phi_{i,j+1} - \phi_{i+1,j+1}}{2\pi}\right) + round\left(\frac{\phi_{ij} - \phi_{i,j+1}}{2\pi}\right) \tag{2.3.2}$$

Each term in (2.3.2) describes the relationship between adjacent pixels in the 2x2 loop under consideration. Each term has a value of 0 if no discontinuity exists between the pixels, $+1$ if there is a positive discontinuity, and -1 if there is a negative discontinuity. s_{ij} may therefore have a value of 0, $+1$, or -1. When s_{ij} has a value of ± 1, the 2x2 loop violates the assumption of the phase field being conservative and the pixel at (i,j) is described as a residue with a "charge" of $+1$ or -1. Refer to Figure 2.7 for examples of positive and negative residues.

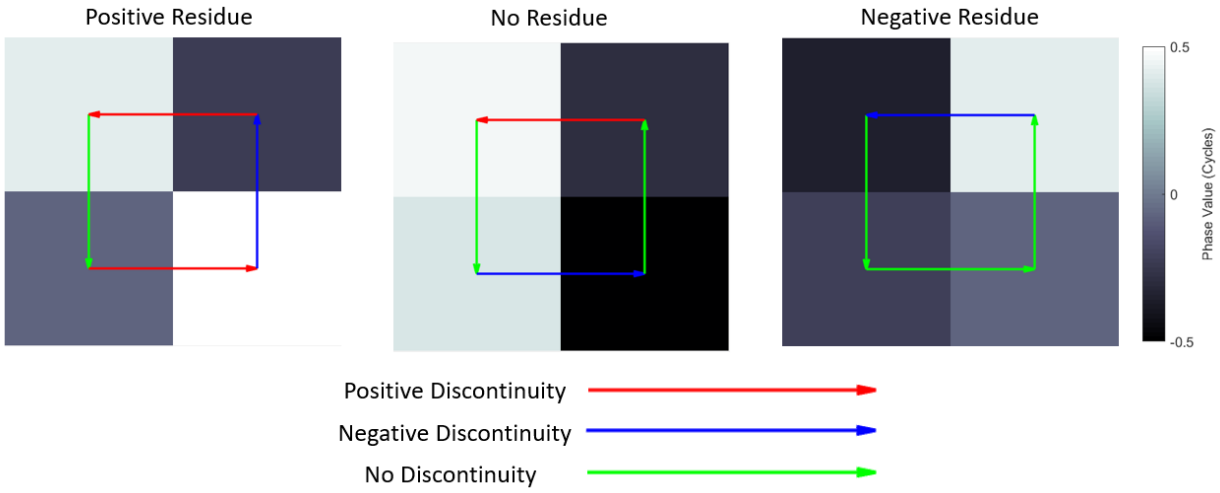

Figure 2.7: Examples of a positive residue (left), a pixel that is not a residue (middle), and a negative residue (right). Positive discontinuities are shown with red vectors, negative discontinuities are shown with blue vectors, and connections with no discontinuity are shown with green vectors. The upper-left pixel corresponds to ϕ_{ij} in (2.3.2).

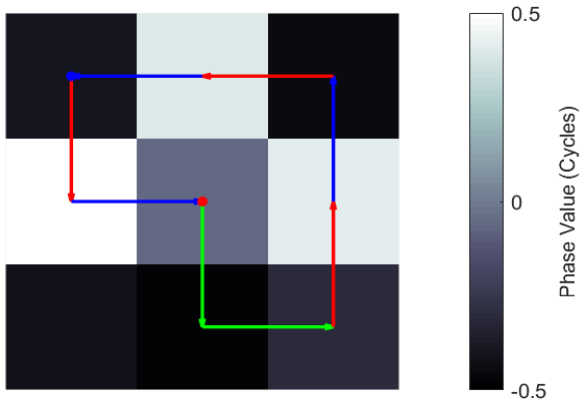

Figure 2.8: An example showing the net discontinuity about a loop containing both a positive residue (red point) and negative residue (blue point).

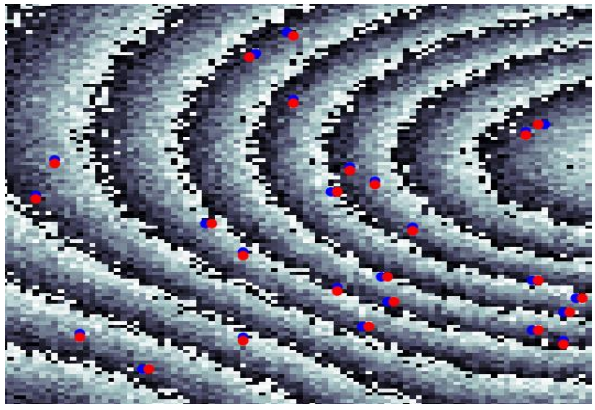

Figure 2.9: A portion of the wrapped signal, with positive residues shown as red points and negative residues shown as blue points.

The total discontinuity about any closed loop may be expressed as a sum of elements of s. If positive and negative residues are joined such that the total charge of the joined residues is 0, then any closed loop containing all of the joined residues will have a net discontinuity of 0 (an example of this is illustrated in Figure 2.8). Each line formed by joining residues is called a "residue cut" line, and if the integration path does not cross any residue cut lines, then the unwrapped signal avoids potential inconsistencies introduced by the residues. A binary mask is created to identify pixels intersecting with residue cut lines, and these pixels will not be visited during the numerical integration. Refer to Figure 2.9 for an example of how residues may be distributed in an interferogram.

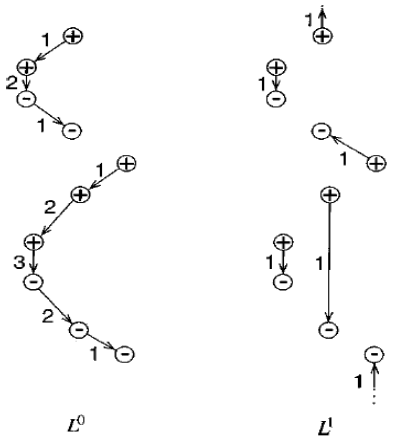

Figure 2.10: Multiple configurations of residue cuts corresponding to the same residue locations/values [Chen and Zebker 2000]. The residue cuts shown in the left image correspond to L-0 norm minimization and the right image shows the results of forming residue cuts using Huntley's [1989] method.

Unwrapping is performed using a flood-fill integration and assigning an arbitrary unwrapped phase to the integration starting point. This method uses (2.3.1) to determine the number of cycles separating neighbouring pixels and unwraps them accordingly. Either a 4-neighbour or 8-neighbour flood-fill method may be used, but there is little practical difference between the two methods; residue cut lines tend to be small and do not greatly restrict the integration path.

The BCPU method is firmly grounded in theory, but in practice it can be difficult to determine how residues should be connected to form residue cuts (this issue is illustrated in Figure 2.10). In Huntley's [1989] documentation, he suggested pairing each positive residue with the nearest negative residue, or with the border of the wrapped signal if it is nearer than the closest negative residue.

Many alternative methods for forming residue cuts have also been proposed. [Chen and Zebker 2000] suggested forming residue cuts based on Minimum Spanning Trees (MST) of residue points. [Karout et al 2006] suggested forming residue cuts using "residue vectors" that describe the directionality associated with each residue. Most popular methods for forming residue cuts seek to

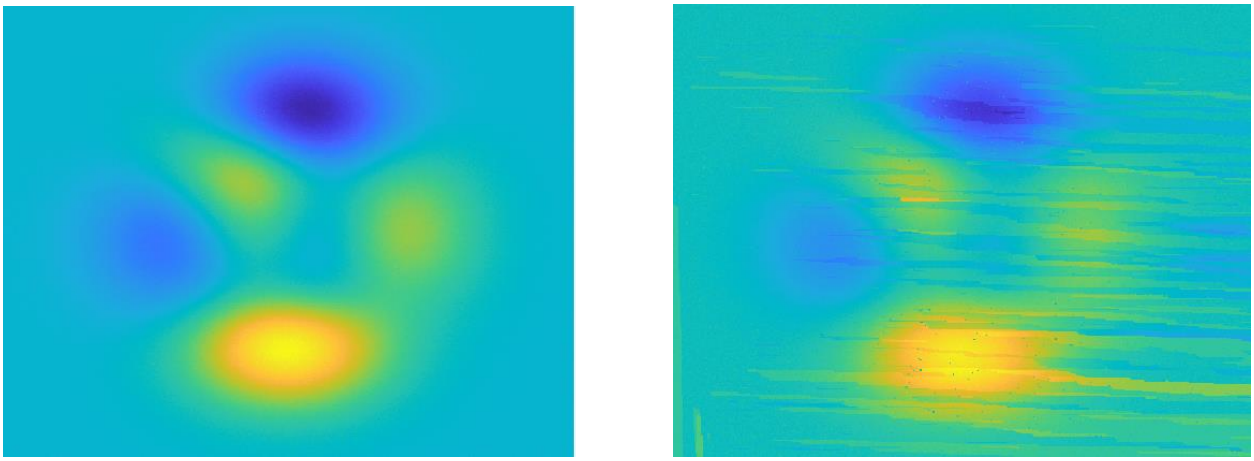

Figure 2.11: The results of applying Huntley's BCPU algorithm to the wrapped signal in Figure 1 (left) and the results of applying the algorithm after additional noise has been injected into the wrapped signal (right). Note the streaked patterns in the right image.

minimize the overall cut length [Gutmann 1999, Zheng 2011]. The formation of residue cuts is an ambiguous process, and many researchers have refined this process in an effort to approximate the L-0 norm minimization of (2.2.3) [Chen and Zebker 2000].

Improperly formed residue cut lines can lead to unwrapping errors. Due to the nature of the unwrapping process, unwrapping errors propagate through the algorithm and form distinct streaking patterns along the integration path. This is clearly illustrated in Figure 2.11, where a streaky pattern similar to that of Figure 2.6 may be seen. It is important to note that the streaks in Figure 2.11 may be traced back to single pixel origins, and that these individual points could be manually added to the automatically-generated residue cut lines to improve results.

The ambiguity in the formation of branch cuts also means that the starting point of the integration has a significant impact on the unwrapping solution, and drastically different unwrapping solutions can be produced by starting the integration at different points. These differences are highlighted in Figure 2.12.

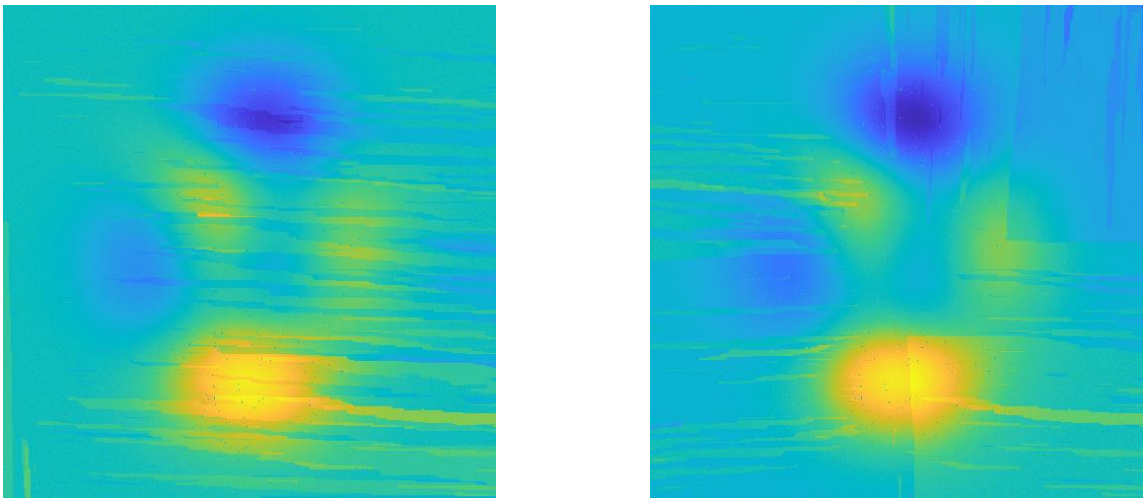

Figure 2.12: Comparison of unwrapping results using the same set of branch cuts but different starting points for the integration. The unwrapped signal on the left used a starting point in the upper-left corner of the interferogram and the unwrapped signal on the right used a starting point in the centre of the interferogram.

2.3.2 Minimum Weighted Discontinuity Phase Unwrapping

The Minimum Weighted Discontinuity (MWD) approach to phase unwrapping was proposed by Flynn [1997], and relies on iterative partitioning of an initial estimation of the unwrapped signal to perform phase unwrapping. Prior to this work, edge detection was used to isolate discontinuous regions in the interferogram and multiples of 2π could then be added to these regions to resolve the discontinuities between them [Lin 1994]. The speckle noise in SAR interferograms make edge detection

unsuitable for resolving these discontinuous regions, and it was this issue that was addressed by the MWD approach [Flynn 1997].

The MWD approach iteratively partitions an interferogram by extending paths along discontinuity curves to form closed loops in the interferogram, and then resolving the discontinuity between the partitioned region and the rest of the interferogram [Flynn 1997]. This process terminates when the paths along the discontinuity curves can no longer be extended to form a closed loop [Flynn 1997].

2.3.3 Region-Growing Phase Unwrapping (RGPU)

Xu [1999] proposed the Region-Growing Phase Unwrapping (RGPU) algorithm, which differs from the BCPU algorithm in that the algorithm starts from multiple seed points and the integration path is determined on-the-fly as opposed to in pre-processing. This approach also uses patch-based phase prediction, as opposed to the pixel-based phase prediction of the branch-cut method.

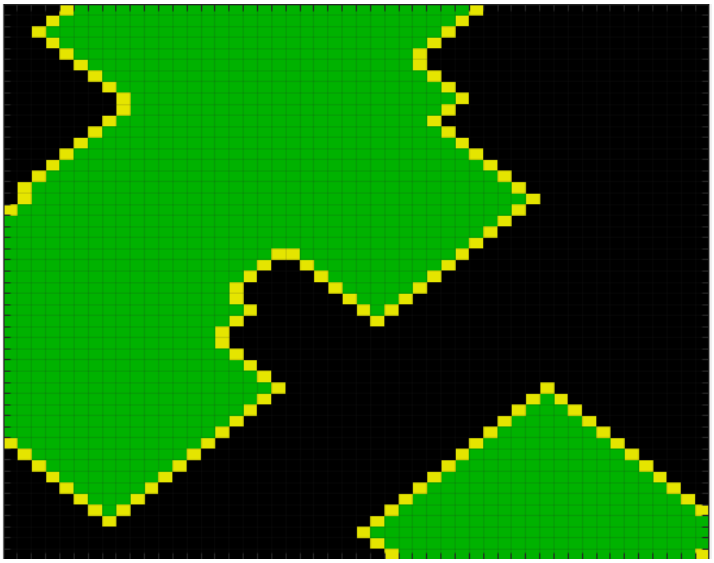

Figure 2.13: Example of the flood-fill based seeded region growing used in the RGPU method. Seeded regions are represented with green cells, and yellow cells represent pixels being unwrapped.

The RGPU algorithm begins with the selection of multiple seed points that will yield reliable solutions in their respective neighbourhoods. Xu [1999] suggested that local variance or coherence may be used to select seed points. Unwrapping is then iteratively performed at the edges of these seeded regions and as the seeded regions begin to overlap, discontinuities between them are resolved and the overlapping regions are merged together. The flood-fill based path definition is used fairly commonly among integration based phase unwrapping algorithms [Huntley 1989, Li et al 2018].

The RGPU algorithm uses patches of unwrapped pixels to predict the phase for wrapped pixels. The number of cycles separating the predicted phase and the wrapped phase is then determined using the equation

$$m = round\left(\frac{\psi_{ij}^{P} - \phi_{ij}}{2\pi}\right) \tag{2.3.3}$$

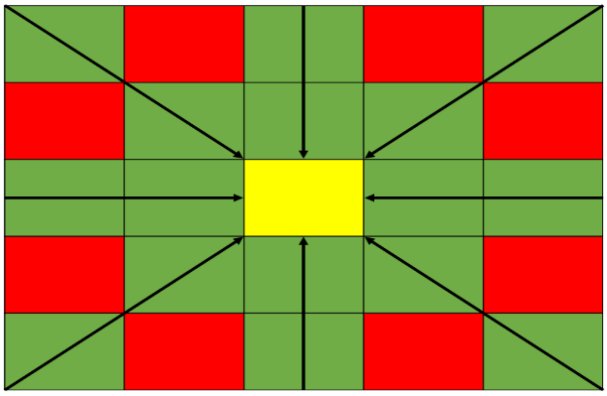

Figure 2.14: Depiction of the prediction lines around a pixel being unwrapped. The yellow cell represents the pixel being unwrapped, the green cells represent pixels in the neighbourhood that can factor into phase prediction, and the red cells represent pixels that cannot factor into phase prediction.

where ψ_{ij}^{P} denotes the predicted phase value for ϕ_{ij}. (2.3.3) is similar to (2.3.1), and is used to unwrap pixels in exactly the same way.

The RGPU algorithm introduced the use of patch-based phase estimation within an integration-based phase unwrapping approach. Patch-based phase estimation has the advantage of being less sensitive to interferometric noise than the pixel-based phase estimation used in the BCPU algorithm. The predicted phase is calculated as the weighted mean of linear extrapolation along prediction lines in the eight directions around the pixel being unwrapped (see Figure 2.14). In the phase prediction stage, each prediction line is checked to determine the number of unwrapped pixels along it. The following cases are defined in Xu's [1999] documentation:

1) If both pixels along the prediction line have been unwrapped, the direction is assigned a weight of 1 and linear prediction is performed using the equation

$$\psi_{k}^{P} = 2\psi_{k} - \psi_{k'} \tag{2.3.4}$$

where ψ_{k}^{P} is the phase prediction for prediction line k, ψ_{k} is the unwrapped phase of the adjacent pixel along the prediction line, and $\psi_{k'}$ is the unwrapped phase of the second pixel along the prediction line.

2) If only the adjacent pixel along the prediction line has been unwrapped, the direction is assigned a weight of 0.5 and zeroth-order prediction is performed using the equation

$$\psi_{k}^{P} = \psi_{k} \tag{2.3.5}$$

3) If the adjacent pixel along the prediction line has not been unwrapped, the direction is assigned a weight of zero and no phase prediction is preformed for that prediction line.

Once the phase has been predicted along each prediction line, they may be combined to form a composite prediction using the weighted mean of the individual predicted phase values using the equation

$$\psi^P = \frac{\sum_k w_k \psi_k^P}{\sum_k w_k} \tag{2.3.6}$$

where w_k denotes the weight associated with prediction line k [Xu 1999].

One of the distinguishing characteristics of the RGPU method is its on-the-fly determination of the integration path. This is accomplished by assessing the unwrapping quality as pixels are being unwrapped and excluding low-quality pixels from future iterations of the algorithm. There are several metrics that Xu [1999] proposed for characterizing the quality of unwrapping:

1) Ensuring that the difference between the predicted phase and the unwrapped phase is lower than a threshold value:

$$|\psi - \psi^P| \leq t_u \tag{2.3.7}$$

2) Ensuring that the deviation between predicted phases from different prediction lines is below a threshold value:

$$\frac{\sum_k w_k |\psi_k^P - \psi^P|}{\sum_k w_k} \leq t_p \tag{2.3.8}$$

3) Ensuring that the local coherence of the interferogram is above a threshold value. This is equivalent to masking low coherence regions in the interferogram.

Any combination of these three measures may be used to assess the quality of unwrapping for a pixel. Xu [1999] suggested jointly applying the constraints presented in (2.3.7) and (2.3.8), and incorporating coherence information if it is available.

The quality measures can be applied adaptively by either modifying the threshold parameters in accordance with local noise levels/coherence in the interferogram or increasing the threshold parameters to become less stringent as the algorithm progresses [Xu 1999].

Unfortunately, these quality control measures cannot be tied to statistical significance levels, and the thresholds must therefore be determined empirically. The appropriate thresholds can be specific to the noise levels/patterns in the interferogram, so the RGPU method can be unreliable if an inappropriate threshold is selected. This also makes it difficult to have the RGPU algorithm adapt to local noise levels in the interferogram.

Another disadvantage of the RGPU method is illustrated in Figure 2.14. When performing phase prediction for a pixel being unwrapped, not all information in the neighbourhood is being used (the red cells in Figure 2.13 indicate pixels that do not inform the phase prediction process). Moreover, the horizontal/vertical prediction lines receive the same weight as the diagonal prediction lines, despite the fact that the diagonal prediction lines are clearly longer (by a factor of $\sqrt{2}$).

2.3.4 SNAPHU

Chen and Zebker's [2000] work forms the basis of the commonly-used SNAPHU algorithms, which draw from branch-cut and MWD phase unwrapping approaches. SNAPHU consists of two algorithms: one that builds upon branch-cut methods and one that builds upon Minimum Cost Flow (MCF) methods [Chen and Zebker 2000]. The branch-cut SNAPHU approach addresses the intractability of L-0 norm minimization by framing the generation of residue cut lines as a minimum rectilinear Steiner tree problem, and then approximating this NP-hard problem using a Minimum Spanning Tree (MST) [Chen and Zebker 2000]. This approach presents a significant improvement over traditional branch-cut approaches and produces much more consistent sets of branch cuts than other approaches, but still loses accuracy when residue points are densely packed [Chen and Zebker 2000].

The MCF phase unwrapping algorithm considers the interferogram to be comprised of a network of the 2 x 2 loops used to determine residue points, and then performs L-1 norm minimization in this network to unwrap the interferogram [Chen and Zebker 2000]. The MCF SNAPHU algorithm uses a similarly-defined network, but seeks to perform L-0 norm minimization with it [Chen and Zebker 2000]. The L-1 norm minimization of the MCF algorithm causes flow costs to be linear along the path of integration, which means that the cost of adding flow to an integration path is independent of the total flow costs of that path [Chen and Zebker 2000]. This increases the likelihood that unwrapping errors will propagate along an integration path in this approach. However, using an L-0 norm minimization in the MCF approach makes the cost of adding flow to an integration path dependent upon the total flow cost of that path [Chen and Zebker 2000]. This causes the approach to naturally prioritize low-cost integration paths, which in turn reduces the chance that unwrapping errors will propagate along any path of integration. Approximating L-0 norm minimization requires an iterative application of the network optimization techniques of the MCF phase unwrapping algorithms, and this makes the approach more reliable but also more computationally-intensive [Chen and Zebker 2000].

2.4 Least-Squares Phase Unwrapping

The matrix equation formulation and iterative solution for the Least-Squares Phase Unwrapping (LSPU) method were first proposed by Hunt [1978], and were later expanded to allow for efficient computation using Discrete Cosine Transformations (DCT) by Ghiglia and Romero [1994]. The LSPU method treats the wrapped differences between adjacent wrapped phase values in the interferogram as observations using the model equations

$$\Delta_w^{(r)}\phi_{ij} = \psi_{i,j+1} - \psi_{ij} \tag{2.4.1}$$

and

$$\Delta_w^{(a)}\phi_{ij} = \psi_{i+1,j} - \psi_{ij} \tag{2.4.2}$$

Here, Hunt's [1978] matrix formulation of the least-squares solution is presented, since the resulting iterative equation forms the basis of the DCT method, and the proposed Path-Based Least-Squares Phase Unwrapping algorithm builds upon the matrix equations developed by Hunt.

A vector containing the wrapped phase values in the interferogram may be defined as

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_{\mathbf{1}}^{(r)^T} \\ \vdots \\ \boldsymbol{\phi}_{\boldsymbol{N}}^{(r)^T} \end{bmatrix} \tag{2.4.3}$$

where $\boldsymbol{\phi}_{\boldsymbol{i}}^{(r)}$ denotes the i[th] row of wrapped phase values in the interferogram.

Similarly, a vector of the unwrapped phase values is defined as

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi}_{\mathbf{1}}^{(r)^T} \\ \vdots \\ \boldsymbol{\psi}_{\boldsymbol{N}}^{(r)^T} \end{bmatrix} \tag{2.4.4}$$

where $\boldsymbol{\psi}_{\boldsymbol{i}}^{(r)}$ denotes the i[th] row of unwrapped phase values in the interferogram.

Using this new notation, the column-wise phase difference model described in (2.4.2) may now be rewritten as

$$\boldsymbol{\Delta}_{\boldsymbol{i}}^{(a)}\boldsymbol{\phi}_{\boldsymbol{i}}^{(r)} = \begin{bmatrix} \psi_{2,i} - \psi_{1,i} \\ \vdots \\ \psi_{M,i} - \psi_{M-1,i} \end{bmatrix} \tag{2.4.5}$$

(2.4.5) may be rewritten in matrix form using the matrix equation [Hunt 1978]

$$\boldsymbol{\Delta}_{\boldsymbol{i}}^{(a)}\boldsymbol{\phi}_{\boldsymbol{i}}^{(r)} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{bmatrix} \boldsymbol{\psi}_{\boldsymbol{i}}^{(r)} = \boldsymbol{D}\boldsymbol{\psi}_{\boldsymbol{i}}^{(r)} \tag{2.4.6}$$

If all $\boldsymbol{\Delta}_{\boldsymbol{i}}^{(a)}\boldsymbol{\phi}_{\boldsymbol{i}}^{(r)}$ are concatenated into a vertical "observation" vector, then the system of equations relating this vector to the unwrapped phases in $\boldsymbol{\Psi}$ is given by [Hunt 1978]

$$\boldsymbol{\Delta}^{(a)}\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{1}}^{(a)}\boldsymbol{\phi}_{\mathbf{1}}^{(r)} \\ \vdots \\ \boldsymbol{\Delta}_{\boldsymbol{N-1}}^{(a)}\boldsymbol{\phi}_{\boldsymbol{N-1}}^{(r)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \boldsymbol{D} \end{bmatrix} \boldsymbol{\Psi} \tag{2.4.7}$$

Using a similar approach as is used for the column-wise phase difference model, the row-wise phase difference model may be expressed as [Hunt 1978]

$$\boldsymbol{\Delta}_{\boldsymbol{i}}^{(\boldsymbol{r})}\boldsymbol{\phi}_{\boldsymbol{i}}^{(\boldsymbol{a})} = \boldsymbol{\psi}_{\boldsymbol{i+1}}^{(\boldsymbol{r})} - \boldsymbol{\psi}_{\boldsymbol{i}}^{(\boldsymbol{r})} \tag{2.4.8}$$

where $\boldsymbol{\phi}_{\boldsymbol{i}}^{(\boldsymbol{a})}$ denotes the i[th] column of wrapped phase values in the interferogram.

In turn, if all $\boldsymbol{\Delta}_{\boldsymbol{i}}^{(\boldsymbol{r})}\boldsymbol{\phi}_{\boldsymbol{i}}^{(\boldsymbol{a})}$ are concatenated into a vertical "observation" vector, the system of equations relating this vector to the unwrapped phases in $\boldsymbol{\Psi}$ is given by [Hunt 1978]

$$\boldsymbol{\Delta}^{(\boldsymbol{r})}\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Delta}_{\mathbf{1}}^{(\boldsymbol{r})}\boldsymbol{\phi}_{\mathbf{1}}^{(\boldsymbol{a})} \\ \vdots \\ \boldsymbol{\Delta}_{\boldsymbol{M-1}}^{(\boldsymbol{r})}\boldsymbol{\phi}_{\boldsymbol{M-1}}^{(\boldsymbol{a})} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{I} & -\boldsymbol{I} \end{bmatrix}\boldsymbol{\Psi} \tag{2.4.9}$$

Bringing all of these "observations" together into a single model yields the system of equations [Hunt 1978]

$$\boldsymbol{\Delta\Phi} = \begin{bmatrix} \boldsymbol{\Delta}^{(\boldsymbol{a})}\boldsymbol{\Phi} \\ \boldsymbol{\Delta}^{(\boldsymbol{r})}\boldsymbol{\Phi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{D} \\ \boldsymbol{I} & -\boldsymbol{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{I} & -\boldsymbol{I} \end{bmatrix}\boldsymbol{\Psi} = \boldsymbol{P\Psi} \tag{2.4.10}$$

The solution for a simple unweighted linear least-squares adjustment may be used to resolve (2.4.10) to solve for the least-squares estimate of $\boldsymbol{\Psi}$:

$$\boldsymbol{P^T P\widehat{\Psi}} = \boldsymbol{P^T\Delta\Phi} \tag{2.4.11}$$

This solution is very impractical for unwrapping most interferograms – most interferograms contain millions of unknown unwrapped phases and the matrix form of the least-squares equation is ill-suited to resolving such a large number of unknowns. To counter this issue, Hunt [1978] showed that the matrix formulation of the least-squares solution is equivalent to the equation

$$\begin{aligned} &\left(\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}\right) + \left(\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}\right) \\ &\quad = \Delta_w^{(a)}\phi_{i,j} - \Delta_w^{(a)}\phi_{i-1,j} + \Delta_w^{(r)}\phi_{i,j} - \Delta_w^{(r)}\phi_{i,j-1} \end{aligned} \tag{2.4.12}$$

(2.4.12) can be applied iteratively to refine estimates of the unwrapped phase values, but cannot be applied if a reasonable initial approximation for the unwrapped phase values is unavailable. Ghiglia and Romero [1994] noted that (2.4.12) is a discretization of Poisson's equation, and can be considered an approximation of the differential equation

$$\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = \rho \tag{2.4.13}$$

where $\rho_{ij} = \Delta_w^{(a)}\phi_{i,j} - \Delta_w^{(a)}\phi_{i-1,j} + \Delta_w^{(r)}\phi_{i,j} - \Delta_w^{(r)}\phi_{i,j-1}$ and is defined everywhere in the interferogram.

The differential equation in (2.4.13) may be solved in the spectral domain so that the unwrapping process can be carried out using convolution operations, which drastically reduces the

computational load of applying the least-squares models. Ghiglia and Romero [1994] used the DCT to accomplish this. The DCT is similar to the Fourier transformation, and is commonly used in image processing and compression. The forward DCT determines the spectral characteristics of a signal f, denoted as f' [Ghiglia and Romero 1994]:

$$f'_{mn} = \sum_i \sum_j 4 f_{ij} \cos\left(\frac{\pi}{2M} m(2i+1)\right) \cos\left(\frac{\pi}{2N} n(2j+1)\right) \tag{2.4.14}$$

The inverse DCT determines the signal f from its spectral characteristics f' [Ghiglia and Romero 1994]:

$$f_{ij} = \frac{1}{MN} \sum_m \sum_n w(m) w(n) f'_{mn} \cos\left(\frac{\pi}{2M} m(2i+1)\right) \cos\left(\frac{\pi}{2N} n(2j+1)\right) \tag{2.4.15}$$

where $w(x)$ is a weighting function that accounts for edge effects in the DCT, and is defined as

$$w(x) = \begin{cases} 1/2, & x = 0 \\ 1 & otherwise \end{cases} \tag{2.4.16}$$

Using the DCT, (2.4.13) may now be used to determine the spectral representation of ψ. Ghiglia and Romero [1994] determined this to be

$$\psi'_{mn} = \frac{\rho'_{mn}}{2\left(\cos\frac{\pi m}{M} + \cos\frac{\pi n}{N} - 2\right)} \tag{2.4.17}$$

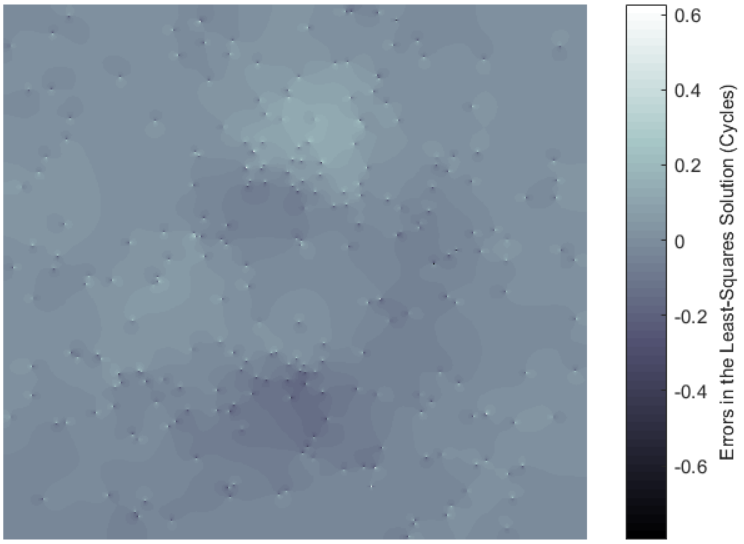

Figure 2.15: Errors in the LSPU solution for the wrapped signal presented in Figure 1.

The unwrapped interferogram may then be determined by performing an inverse DCT on ψ'.

It is important to recognize that, since the LSPU method is a global optimization method, the unwrapped phase values will not exactly satisfy (2.1.1) as they would for the BCPU and RGPU methods. The random error in the wrapped phase gets distributed among neighbouring pixels by the LSPU process.

When analyzing the differences between the LSPU solution and the simulated signal (see Figure 2.15), it is apparent that there are distinct, large-scale patterns to the errors in the LSPU solution. In Figure 2.15, the errors in the LSPU solution has two characteristics:

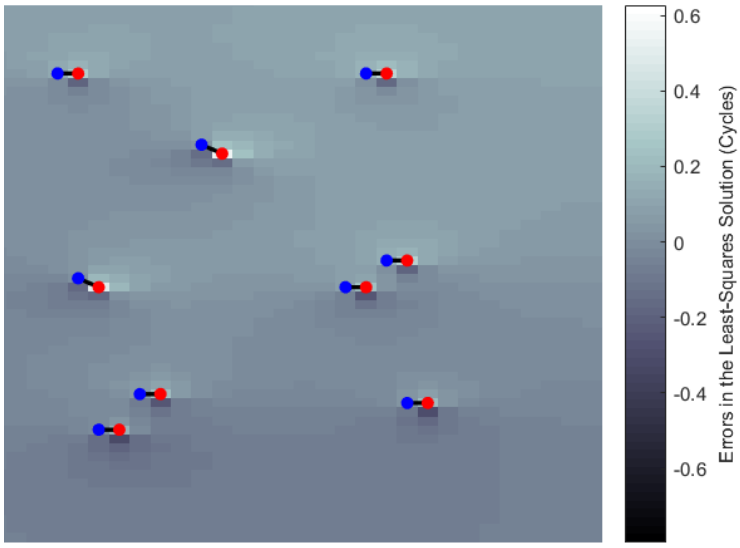

Figure 2.16: A portion of the LSPU errors shown in Figure 13, with residue and residue cut information superimposed. Positive and negative residues are red and blue circles, respectively, and residue cuts are black lines connecting positive and negative residues.

1) Broad regions have very similar errors, with distinct boundaries between regions; and

2) There are large, dipole-like positive/negative error pairings that cross residue cut lines (highlighted in Figure 2.16).

Both of these phenomena can be explained by the fact that the LSPU algorithm includes phase differences that cross residue cut lines (the residue cut algorithms were developed to counter this very issue, albeit for integration-based unwrapping). These phase differences are artificially large due to random noise in the interferogram, and they have a pronounced impact on the least-squares solution.

While the impact of the residues is less extreme than for the BCPU algorithm, residues still contribute to an overall disagreement between the least-squares solution and the "ideal" solution, with larger errors concentrated around residues.

2.4.1 The Weighted Least-Squares Approach

Applying weights in the LSPU approach is equivalent to solving the matrix equation [Ghiglia and Romero 1994]

$$\boldsymbol{WP\hat{\Psi}} = \boldsymbol{W\Delta\Phi} \tag{2.4.18}$$

where $\boldsymbol{W}$ denotes a weighting matrix. These weights may be determined using local coherence or noise values in an interferogram, or they may be manually defined for certain regions in an interferogram.

The normal system of equations for the system of equations in (2.4.18) is then

$$\boldsymbol{P^T W^T W P\hat{\Psi}} = \boldsymbol{P^T W^T W\Delta\Phi} \tag{2.4.19}$$

If $\boldsymbol{Q} = \boldsymbol{P^T W^T W P}$ and $\mathbf{c} = \boldsymbol{P^T W^T W\Delta\Phi}$, then (2.4.19) is rewritten as [Ghiglia and Romero 1994]

$$\boldsymbol{Q\hat{\Psi}} = \boldsymbol{c} \tag{2.4.20}$$

The quantity $\boldsymbol{Q}$ may be considered the sum of the matrix $\boldsymbol{P^T P}$ and a difference matrix $\boldsymbol{S}$ such that $\boldsymbol{Q} = \boldsymbol{P^T P} + \boldsymbol{S}$. Substituting this definition of $\boldsymbol{Q}$ into (2.4.20) produces the equation [Ghiglia and Romero 1994]

$$\boldsymbol{P^T P\hat{\Psi}} = \boldsymbol{c} - \boldsymbol{S\hat{\Psi}} \tag{2.4.21}$$

Here, the left side of (2.4.21) is identical to the left side of the (2.4.11), which provides the normal equations for the unweighted case. This suggests that the DCT-based solution to the least-squares equations may be applicable to the weighted case as well. This is complicated by the appearance of the unknown $\boldsymbol{\hat{\Psi}}$ on the right side of (2.4.21). To resolve a weighted estimate for $\boldsymbol{\hat{\Psi}}$, (2.4.21) must be applied iteratively as [Ghiglia and Romero 1994]

$$\boldsymbol{P^T P\hat{\Psi}_{k+1}} = \boldsymbol{c} - \boldsymbol{S\hat{\Psi}_k} \tag{2.4.22}$$

The vector $\boldsymbol{c}$ can be estimated as long as the weights being used are known, and remains constant from one iteration to the next. The matrix $\boldsymbol{S}$ can be used alongside the previous estimate of the unwrapped phase to resolve the vector on the right side of (2.4.22). Once this is accomplished, the DCT approach to least-squares phase unwrapping may be directly applied to resolve an estimate for $\boldsymbol{\hat{\Psi}_{k+1}}$. This process is applied iteratively until either a maximum number of iterations is reached or until the magnitude of $\boldsymbol{S\hat{\Psi}_k}$ is sufficiently small.

3 Proposed Phase Unwrapping Methods

The two proposed phase unwrapping methods are developed upon existing phase unwrapping algorithms, with a mind to share their strengths and mitigate their weaknesses. Two significant innovations have been developed through this research:

1) The Polynomial-Based Region-Growing Phase Unwrapping (PBRGPU) method, which is structured in the same manner as the Region-Growing Phase Unwrapping (RGPU) method but utilizes least-squares polynomial fitting for phase-prediction. This not only results in a more reliable predicted phase, but also allows for rigorous on-the-fly statistical tests to assess the quality of the unwrapping solution for a given pixel.
2) The Path-Based Least-Squares Phase Unwrapping (PBLSPU) method, which exploits the measurement model applied in Least-Squares Phase Unwrapping (LSPU), performing global optimization using small neighbourhoods along an integration path. The strategic objective of this method is to produce a globally optimized solution, and allow for ongoing, statistically rigorous removal of high-noise measurements as well. This could not be accomplished by using the DCT to resolve the least-squares solution.

Both methods have been designed to share the advantages of the methods they are based on, but also to mitigate the disadvantages of the pre-existing methods. In particular, the PBLSPU method seeks to combine the advantages of both LSPU methods and path-based integration phase unwrapping methods.

3.1 Polynomial-Based Region-Growing Phase Unwrapping (PBRGPU)

The PBRGPU algorithm utilizes the same processes of integration path determination and seeded region merging as the RGPU method proposed by Xu and Cumming [1999]. The main difference between the PBRGPU and the RGPU methods lies in the phase estimation processing step. The PBRGPU method utilizes a least-squares polynomial fitting to perform phase prediction, rather than defining phase prediction as the weighted sum of multiple linear predictions. This allows for an estimation of the error associated with the predicted phase, which in turn enables rigorous statistical testing of the predicted/unwrapped phase for each pixel.

Just as with the RGPU method, the PBRGPU algorithm starts from multiple seed points which are selected to yield reliable solutions in their respective neighbourhoods. As in Xu [1999], local

interferogram variance or coherence may be used as a measure of this reliability. Unwrapping is still performed at the edges of the seeded regions, and discontinuities between regions are resolved once they begin to overlap.

Upon visiting a pixel at the edge of an unwrapped seeded region, the PBRGPU algorithm uses all unwrapped pixels in the 5 x 5 neighbourhood around the pixel being unwrapped to predict its unwrapped phase using a gridded bivariate least-squares polynomial fitting. A second order model is preferred, but for neighbourhoods with few unwrapped pixels, a first or zeroth order model may need to be used instead. The second order polynomial model is given by the equation

$$\psi_{kl} = a_0 + a_1 k + a_2 l + a_3 k^2 + a_4 kl + a_5 l^2 + \xi_{kl} \tag{3.1.1}$$

where ψ_{kl} denotes the unwrapped phase in the 5 x 5 neighbourhood about the pixel being unwrapped and both k and l range from -2 to 2 (defining the pixel being unwrapped as being at the centre of the neighbourhood). ξ_{kl} is a random noise term, which is assumed to follow a normal distribution with zero mean and a variance of σ_ϕ^2. Note that this only assumes that the noise in the interferogram follows a normal distribution locally within each processing patch.

Since the pixel being unwrapped is defined to be at the centre of the neighbourhood, the predicted value for its unwrapped phase is simply a_0. The least-squares estimate of a_0 may be determined alongside the other polynomial coefficients from the unweighted linear parametric least-squares solution

$$\hat{\boldsymbol{a}} = \left(\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{l} \tag{3.1.2}$$

wherein $\boldsymbol{A}$ is the first design matrix, defined as

$$\boldsymbol{A} = \begin{bmatrix} 1 & k_1 & l_1 & k_1^2 & k_1 l_1 & l_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & k_N & l_N & k_N^2 & k_N l_N & l_N^2 \end{bmatrix} \tag{3.1.3}$$

and $\boldsymbol{l}$ is the vector of observations, defined as

$$\boldsymbol{l} = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_N \end{bmatrix} \tag{3.1.4}$$

The *a priori* covariance matrix of $\hat{\boldsymbol{a}}$ is given by

$$\boldsymbol{C}_{\hat{\boldsymbol{a}}} = \left(\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A}\right)^{-1} \tag{3.1.5}$$

The *a posteriori* covariance matrix of $\hat{\boldsymbol{a}}$ is $\boldsymbol{C}_{\hat{\boldsymbol{a}}}$ in (3.1.5) scaled by the *a posteriori* variance factor

$$\hat{\sigma}_0^2 = \frac{(\boldsymbol{l} - \boldsymbol{A}\hat{\boldsymbol{a}})^T(\boldsymbol{l} - \boldsymbol{A}\hat{\boldsymbol{a}})}{dof} \tag{3.1.6}$$

where dof denotes the degrees of freedom of the polynomial fitting, such that

$$\widehat{\boldsymbol{C}}_{\hat{\boldsymbol{a}}} = \hat{\sigma}_0^2 \boldsymbol{C}_{\hat{\boldsymbol{a}}} \tag{3.1.7}$$

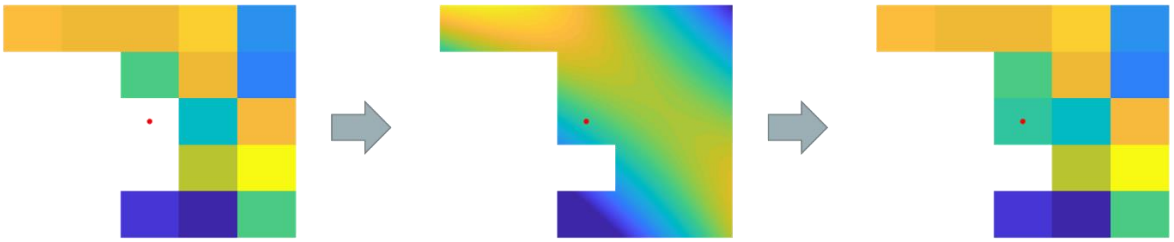

Figure 3.1: Visualization of the polynomial-based phase estimation process for the pixel being unwrapped (marked by the red dot). The unwrapped phase values in the neighbourhood of the pixel are retrieved (left), then a bivariate polynomial model is fit to the unwrapped phases (middle), and this model is used to predict the unwrapped phase for the pixel being unwrapped (right).

The predicted phase is used to unwrap the pixel under consideration using (2.3.3).

Since the predicted phase is simply the first element of $\hat{\boldsymbol{a}}$, the variance of the predicted phase is represented by the first diagonal element of $\widehat{\boldsymbol{C}}_{\hat{\boldsymbol{a}}}$. This allows for the construction of a test statistic to determine the quality of unwrapping. Here, the null hypothesis is given as

$$H_0: \psi_{ij} = \tilde{\psi}_{ij} \tag{3.1.8}$$

with a corresponding test statistic of

$$t_{ij} = \frac{\psi_{ij} - \psi_{ij}^P}{\sqrt{\hat{\sigma}_{\hat{a}_0}^2 + \sigma_0^2}} \tag{3.1.9}$$

where i and j denote the location of the pixel being unwrapped in the interferogram.

This test statistic is very similar in form to (2.3.7), but should follow a t-distribution and therefore may be tied to a specific statistical significance level. Assuming that the unwrapped phase values in the 5 x 5 neighbourhood are normally-distributed data, t_{ij} should follow a *t*-distribution with degrees of freedom equal to the degrees of freedom of the least-squares polynomial fitting.

(2.3.8) used the overall fit of the phase prediction model to the unwrapped phases to characterize the quality of the solution. Similarly, the *a posteriori* variance factor in (3.1.6) is used to construct a test statistic

$$x_{ij}^2 = dof \frac{\hat{\sigma}_0^2}{\sigma_0^2} \tag{3.1.10}$$

under the null hypothesis

$$H_0: \sigma_\psi^2 = \sigma_\phi^2 \tag{3.1.11}$$

The test statistic in (3.1.10) should follow a χ^2 distribution with dof degrees of freedom. In the test statistic, $\hat{\sigma}_0^2$ provides an estimate of the variance of the unwrapped phase, and σ_0^2 provides information about the variance of the wrapped phase.

There is some ambiguity when it comes to defining the *a priori* value for σ_0^2, since it represents the expected noise in the interferogram. Noise levels can vary significantly over an interferogram, and this makes it difficult to determine the specific noise level expected prior to conducting phase unwrapping. This method therefore requires a reliable identification of the *a priori* noise level in order to be effective.

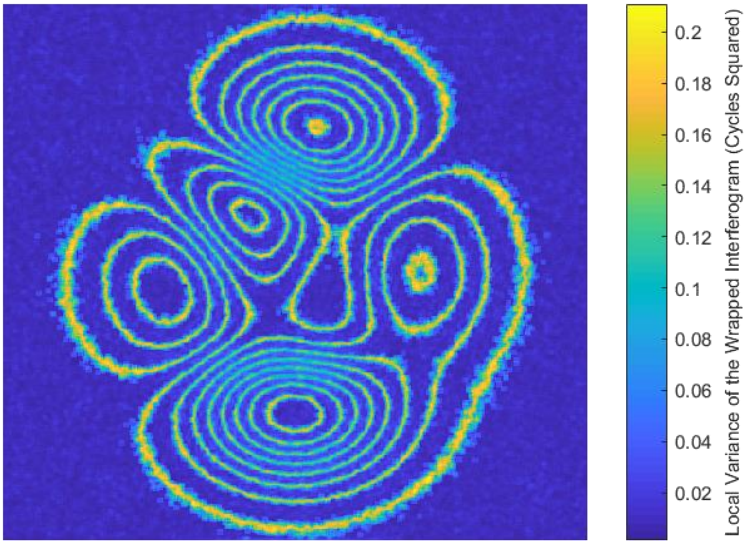

Figure 3.2: Local variance in a simulated wrapped interferogram. Notice the large spikes in variance near discontinuities in the interferogram.

Two approaches to identifying *a priori* noise levels in the interferogram are considered here: determining a constant *a priori* noise level over the entire interferogram, and an adaptive *a priori* noise level locally over the extent of the interferogram. Both approaches use local variances across the interferogram to estimate *a priori* variances.

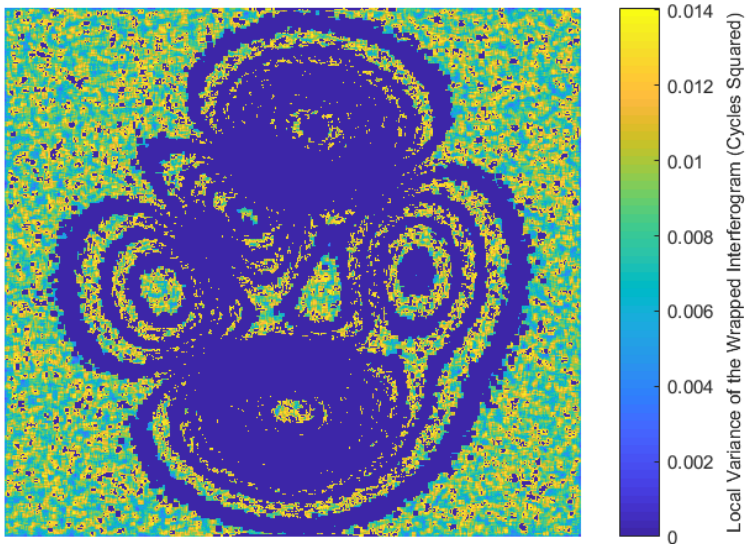

Figure 3.3: Local variance in a simulated wrapped interferogram after thresholding is applied. The thresholded regions are not considered in the estimation of the a priori variance of the wrapped phases.

The constant *a priori* variance approach considers the *a priori* variance of the wrapped phases in the interferogram to be the weighted mean of the local variances across the interferogram. As can be seen in Figure 3.2, there are spikes in the local variance near

discontinuities in the interferogram. These spikes in the local variance are produced artificially by the discontinuities in the wrapped interferogram and are not representative of the noise levels in the interferogram. These values may be removed via thresholding. Figure 3.3 shows the local variances of a simulated signal when the values above $1/15$ of the maximum local variance are removed. The *a priori* variance of the wrapped phases is then calculated as the mean of the remaining local variances after thresholding.

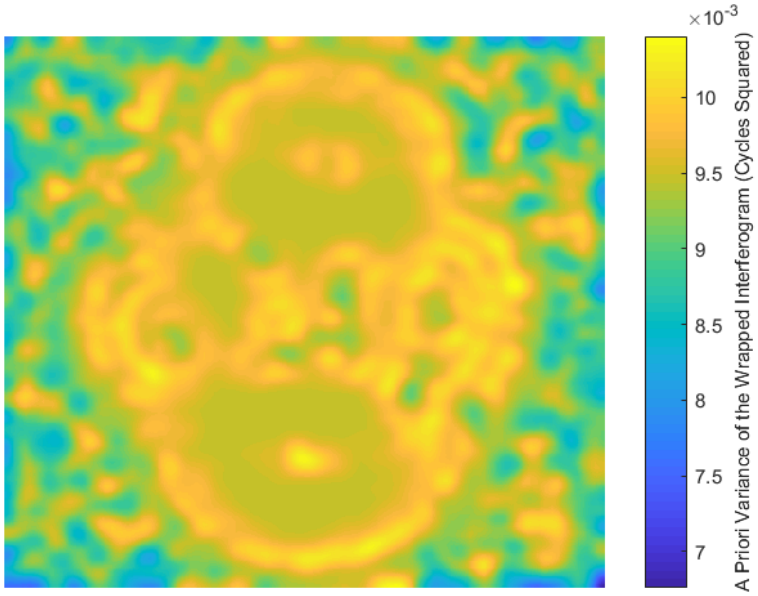

Figure 3.4: A Priori variance of a simulated wrapped interferogram.

The adaptive *a priori* variance approach estimates the moving-window weighted mean of local variances across the interferogram. As with the constant *a priori* variance approach, thresholding is applied and the mean local variance of the remaining pixels is used to replace the values of the removed pixels. The local variances use small neighbourhoods, and so do not provide reliable estimates of the *a priori* variances of the wrapped interferogram. To overcome this issue, a low-pass Gaussian filter is applied to estimate the *a priori* variance that adapts to broader trends in local noise levels in the interferogram. An example of adaptive *a priori* variance values is shown in Figure 3.4.

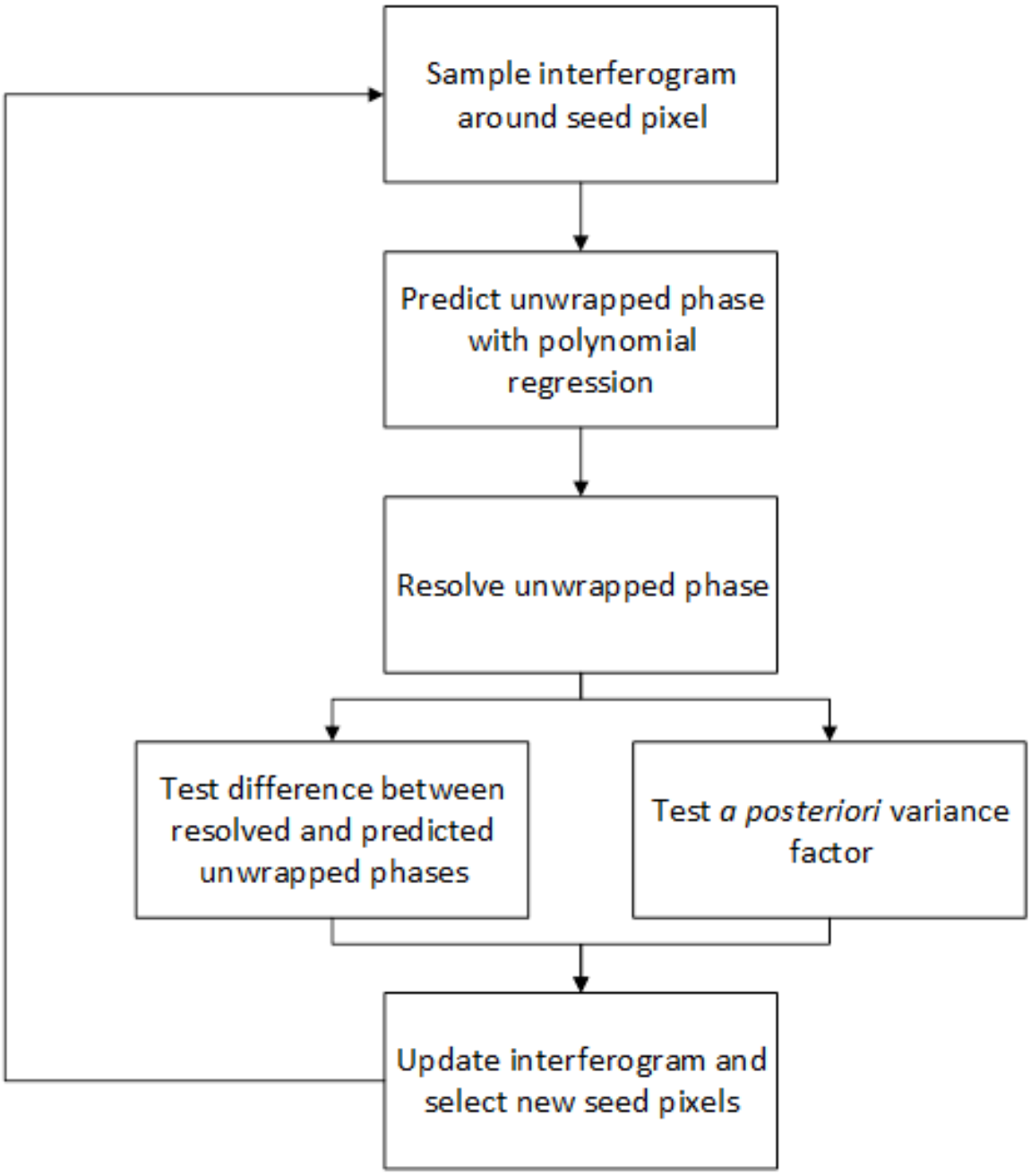

Figure 3.5: Flow chart of phase estimation in the Polynomial-Based Region-Growing Phase Unwrapping (PBRGPU) algorithm.

3.2 Path-Based Least-Squares Phase Unwrapping (PBLSPU)

Although the observation model of the PBLSPU method is identical to that of the LSPU method, this approach deviates from the LSPU algorithm in its path-based implementation of the least-squares model. This is accomplished by expanding the solution about a single seeded region using the method of sequential least-squares.

For this method, any neighbourhood size may be used, with a preference for smaller neighbourhoods that make a direct application of (2.4.11) more feasible, which allows for the complete removal of suspected high-noise measurements. More information about the impact neighbourhood size has on the unwrapping solution is provided in Section 4.2.2.

Of course, ensuring that the solution satisfies the global optimization criterion used in the LSPU algorithm requires that the estimates of unwrapped phases from earlier iterations of the algorithm be incorporated into the measurement model through a sequential least-squares processing approach. This may be accomplished through the inclusion of the pseudo-observations

$$\boldsymbol{\Psi_0} = \mathbf{P_0}\boldsymbol{\Psi} \tag{3.2.1}$$

with the associated observation covariance matrix of $\boldsymbol{C_{\Psi_0}}$. It should be noted that $\boldsymbol{C_{\Psi_0}}$, the covariance matrix of the estimated unwrapped phases, should be determined through earlier iterations of the algorithm. These pseudo-observations are only applied for pixels that already have an estimated unwrapped phase value – the $\boldsymbol{P_0}$ matrix is therefore defined as an identity matrix with the rows corresponding to the pixels that do not have unwrapped phase estimates removed.

In the unweighted case, the observation vector becomes

$$\boldsymbol{l} = \begin{bmatrix} \boldsymbol{\Delta\Phi} \\ \boldsymbol{\Psi_0} \end{bmatrix} \tag{3.2.2}$$

the overall first design matrix then becomes

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{P_0} \end{bmatrix} \tag{3.2.3}$$

with the covariance matrix of the observations

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{C_{\Psi_0}} \end{bmatrix} \tag{3.2.4}$$

The least-squares solution is given as follows

$$\hat{\boldsymbol{\Psi}} = \left(\boldsymbol{P^T P} + \boldsymbol{P_0^T C_{\Psi_0}^{-1} P_0}\right)^{-1}\left(\boldsymbol{P^T \Delta\Phi} + \mathbf{P_0^T}\boldsymbol{C_{\Psi_0}^{-1}}\boldsymbol{\Psi_0}\right) \tag{3.2.5}$$

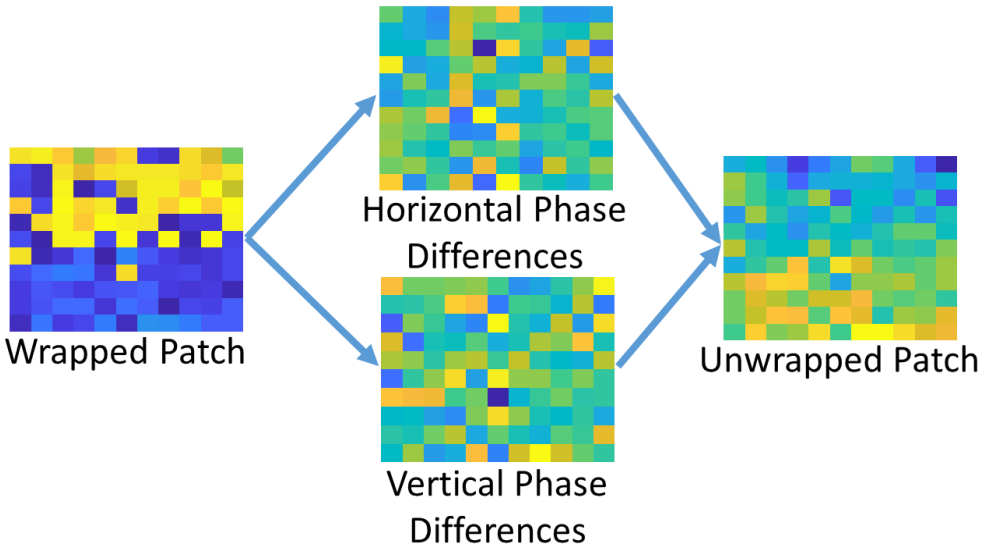

Figure 3.6: Example of the LSPU matrix equations being applied directly for an 11 x 11 patch in a wrapped interferogram. It is important to note that the horizontal and vertical phase differences constitute different observations when applying the matrix equations directly, rather than them combining to form a single quantity as is the case when using DCTs to resolve the least-squares unwrapped phases.

One of the main motivations of this approach is to add the capability of applying least-squares phase unwrapping while removing high-noise observations. This is accomplished *a posteriori* after the variance factor

$$\hat{\sigma}_0^2 = \frac{\left(\boldsymbol{P}\widehat{\boldsymbol{\Psi}} - \boldsymbol{\Delta\Phi}\right)^{\boldsymbol{T}}\left(\boldsymbol{P}\widehat{\boldsymbol{\Psi}} - \boldsymbol{\Delta\Phi}\right) + \left(\boldsymbol{P_0}\widehat{\boldsymbol{\Psi}} - \boldsymbol{\Psi_0}\right)^{\boldsymbol{T}}\boldsymbol{C}_{\boldsymbol{\Psi_0}}^{\boldsymbol{-1}}\left(\boldsymbol{P_0}\widehat{\boldsymbol{\Psi}} - \boldsymbol{\Psi_0}\right)}{dof} \tag{3.2.6}$$

where dof describes the degrees of freedom of the adjustment system. This variance estimation could also be accomplished using variance component estimation to provide on-the-fly refinements to the weighting scheme of the observations. Without applying variance component estimation, this method produces identical results to the unweighted least-squares phase unwrapping solution.

The variance factor is used to determine *a posteriori* estimates of the variances of the residuals using the equation

$$\boldsymbol{C}_{\hat{\boldsymbol{v}}} = \hat{\sigma}_0^2\left(\boldsymbol{C} - \boldsymbol{A}\left(\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{C}^{\boldsymbol{-1}}\boldsymbol{A}\right)^{\boldsymbol{-1}}\boldsymbol{A}^{\boldsymbol{T}}\right) \tag{3.2.7}$$

Knowing the *a posteriori* variance of the residuals corresponding to each observation allows for a rigorous statistical assessment of whether or not each observation has a high noise level. In this case, the null hypothesis for statistical testing would be

$$H_0: \boldsymbol{A}\widetilde{\boldsymbol{\Psi}} = \tilde{\boldsymbol{l}} \tag{3.2.8}$$

with a corresponding set of test statistics defined as

$$t_i = \frac{\hat{v}_i}{\hat{\sigma}_{\hat{v}_i}}, i = 1, \dots, n \tag{3.2.9}$$

where $\hat{v}_i$ denotes the residual for the i[th] observation in the least-squares model. Assuming data in the processing patch to be normally-distributed, the test statistic in (3.2.9) follows a *t*-distribution with degrees of freedom equal to that of the adjustment system.

Since PBLSPU is a path-based approach that applies the least-squares principle in patches along the integration path, the unwrapped signal is more vulnerable to individual high noise pixels in the wrapped signal than in the standard LSPU approach (the degrees of freedom is relatively low for each patch). This makes (3.2.9) insufficient on its own for assessing solution quality, and additional quality assurance should be employed. The statistical test in (3.1.9) provides an intuitive and rigorous additional measure of solution quality. In order to apply this test, the least-squares estimates of the unwrapped phase fulfill the role of ψ_{ij}^P, while ψ_{ij} is defined by (2.3.3). It is important to note that, although the unwrapped phases in each patch are being estimated using the exact model given by (2.3.3), it is still the least-squares estimate of these unwrapped phases that are used to populate the unwrapped interferogram.

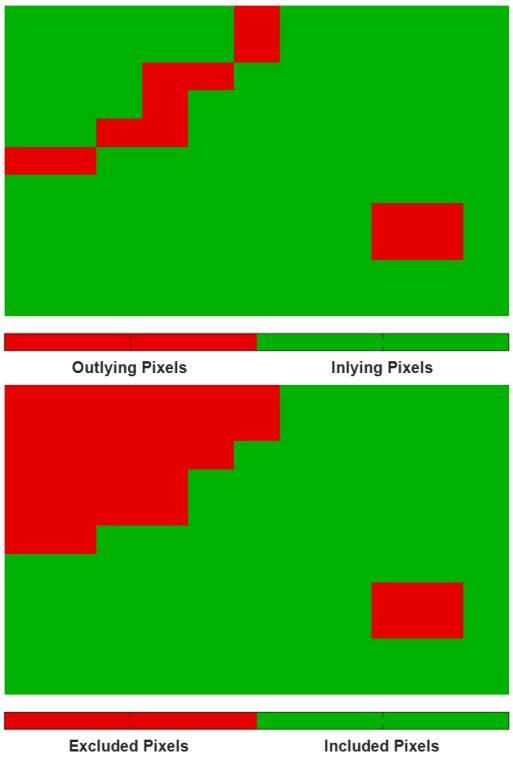

Figure 3.7: Illustration of why the "connectedness" of pixels in the solution is important. The top image shows inlying/outlying pixels, and outlying pixels result in there being a disconnected region in the upper-left corner of the interferogram patch. The bottom image shows the pixels that are included in the least-squares solution. Note that the region in the upper left corner of the patch is excluded.

An additional complication arises from the fact that the path-based least-squares method potentially flags multiple pixels as high-noise pixels in each processing patch. Path-based phase unwrapping algorithms require an uninterrupted integration path to prevent artificial discontinuities from being introduced to the solution. Given the nature of the path-based least-squares method, some pixels may be excluded so as to create multiple disconnected areas in a patch (refer to Figure 3.7). This must be resolved by ensuring that only those pixels connected to the seed pixel are included in the solution at any iteration. This can be realized by

masking high-noise pixels and using flood-fill region-growing to identify all pixels that are connected to the seed pixel.

If the null hypothesis for either statistical test is rejected for an individual pixel or if that pixel is not connected to the seed pixel by other unwrapped pixels, that pixel may be flagged as a pixel to exclude from the unwrapping solution. Once all high-noise pixels in a patch have been flagged, the least-squares estimates of the unwrapped phases may be re-evaluated in order to exclude all high-noise observations. Solution re-evaluation and the detection and removal of high-noise pixels can be applied iteratively until the flagged high-noise pixels remain the same from one iteration to the next, or until the number of iterations reaches a specified maximum. After a final phase unwrapping solution has been produced for a patch, it is important to verify that all unwrapped pixels are connected to the seed pixel via an uninterrupted integration path.

It is important to note that, as with the PBRGPU approach, the assumption of noise being normally distributed is only being made locally for a processing patch. The least-squares models do not require the noise to be normally distributed (as long as there is no bias in the noise), but the statistical quality assurance does rely on this assumption.

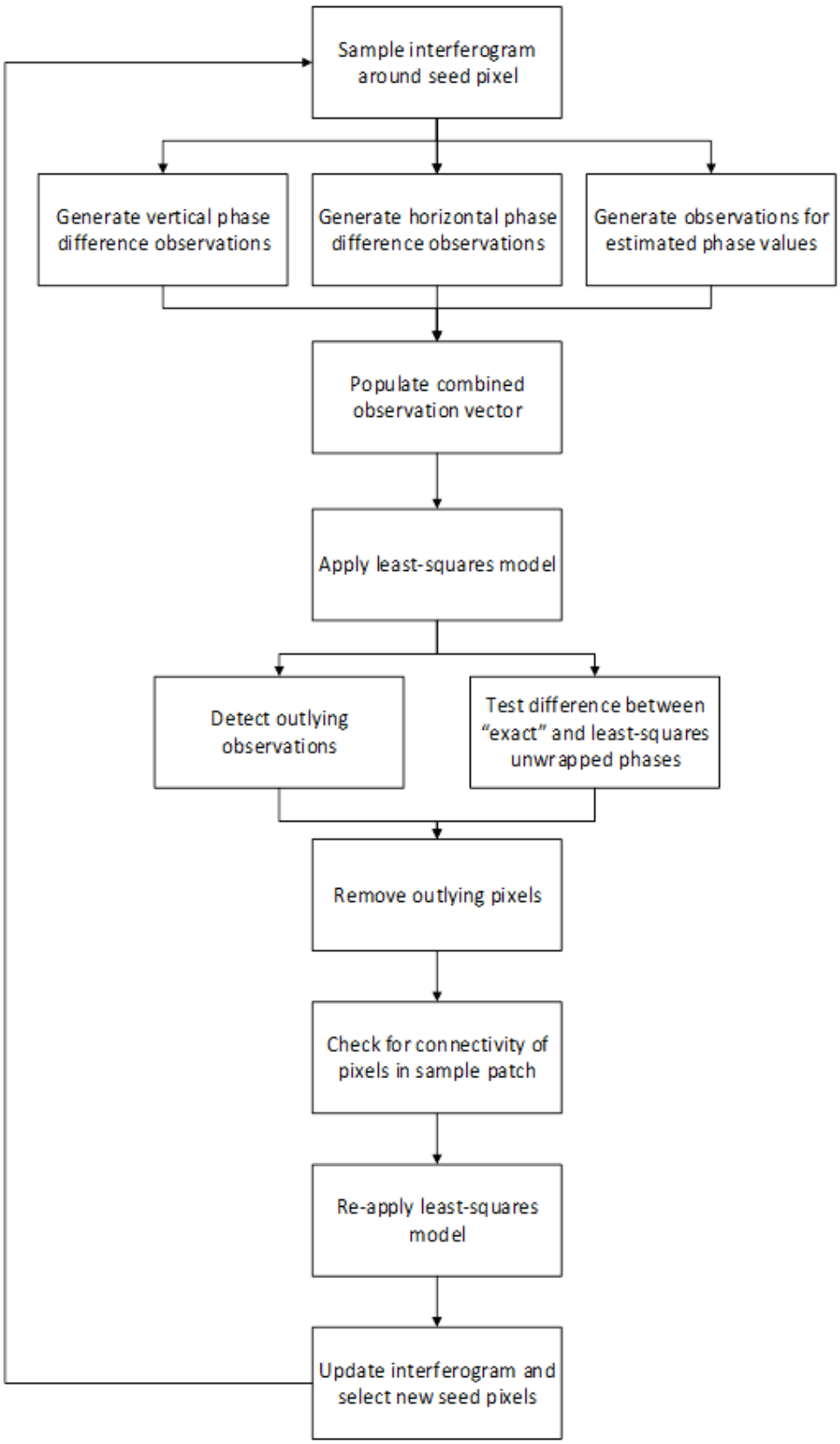

Figure 3.8: Flowchart of phase estimation in the path-based least-squares phase unwrapping algorithm

4 Results

4.1 Simulated Data

The proposed algorithms were first tested using simulated interferograms. This allowed for the definition of an ideal unwrapped signal, which in turn facilitated a direct comparison between the unwrapping quality of different phase unwrapping algorithms. The simulated interferograms were generated using various levels of normally-distributed noise to verify the performance of the proposed algorithms under different noise conditions (definition of the different noise levels used is shown in Table 4.1 and the simulated wrapped signals are shown in Figure 4.1 with the corresponding ideal unwrapped signals in Figure 4.2). The signals were generated by wrapping the Matlab Peaks function and adding normally-distributed noise. All simulated signals were 500 x 500 pixels. It is important to note that these signals are intended to test the phase unwrapping approaches under various noise conditions, as the simulated signals do not simulate the speckle noise present in SAR interferograms.

Three specific measures were used to assess and describe the quality of a phase unwrapping solution for each simulated wrapped signal: the Root Mean Square Error (RMSE) between the unwrapped signal and the ideal solution; the percentage of pixels that were considered high-noise pixels by the phase unwrapping approach (this could not be applied to the LSPU solution, since the detection of high-noise pixels is not possible with this approach); and whether or not there were any observable discontinuities in the unwrapped signal (this also could not be applied to the LSPU solution, since it is defined to be a continuous surface).

The PBRGPU, PBLSPU, LSPU, and RGPU phase unwrapping algorithms were each used to unwrap the simulated wrapped signals, so that the proposed phase unwrapping algorithms could be compared to the approaches that they build upon. When applying the RGPU algorithm, the strictest parameters put forth in Xu [1999] were used ($t_u = \pi/4$, $t_p = \pi/4$).

Table 4.1: Summary of the noise characteristics of the simulated signals used to test the phase unwrapping algorithms. The level of noise is characterized by the standard deviation of normally-distributed random noise that was added to the ideal simulated signal.

Noise Level	Standard Deviation (Percent of a Cycle)
None	0
Low	5
Medium	10
High	15

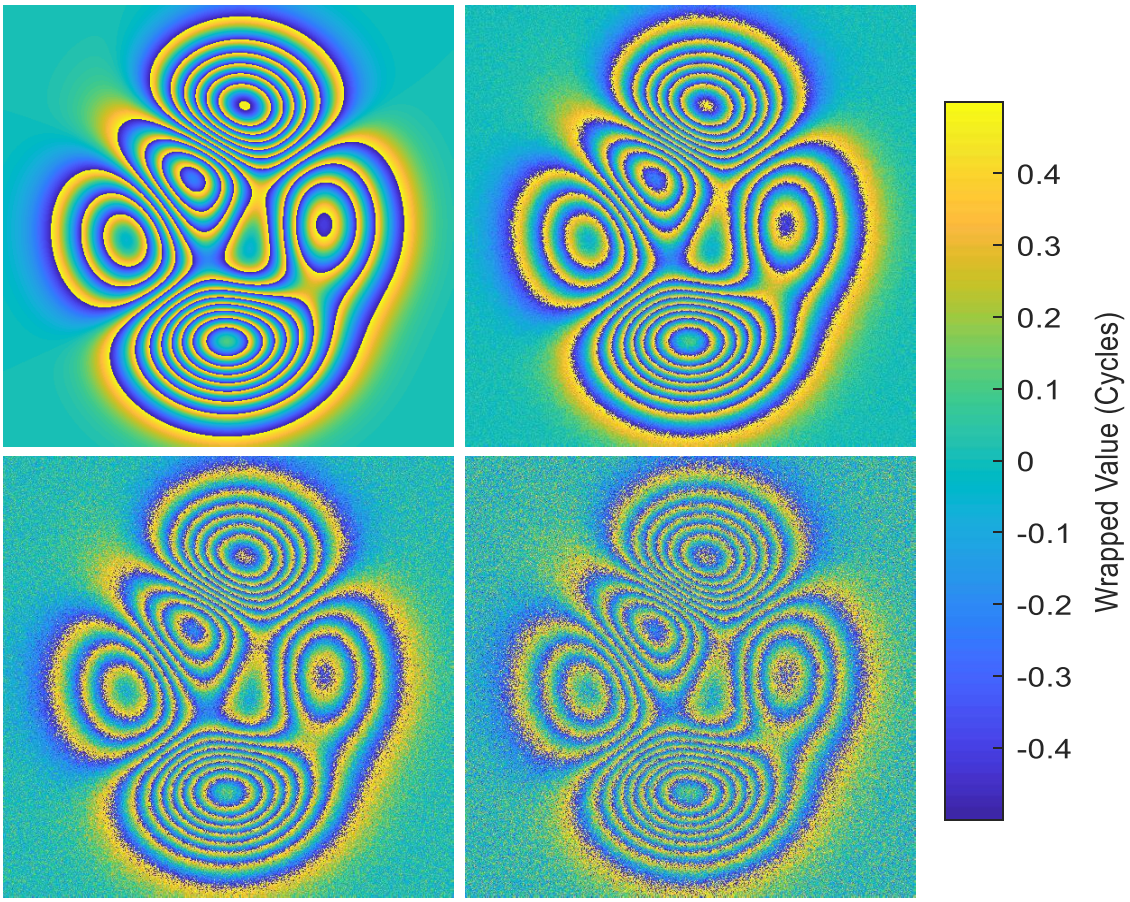

Figure 4.1: The simulated wrapped signals used to test the proposed phase unwrapping algorithms, in cycles. The top-left signal has no noise, the top-right has low noise, the bottom left has medium noise, and the bottom right has high noise.

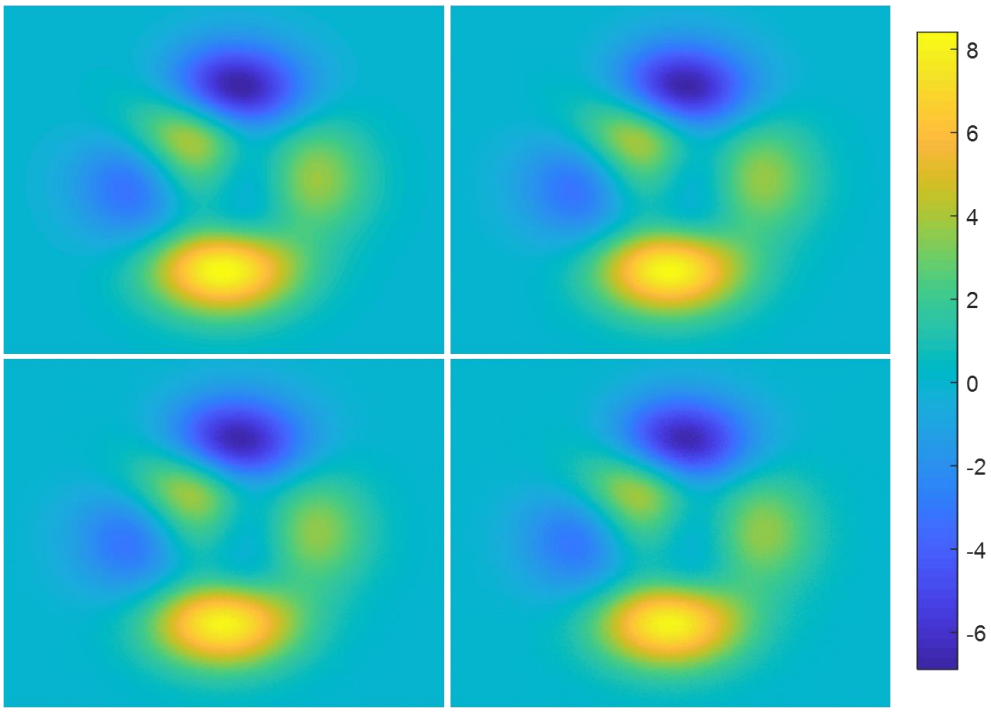

Figure 4.2: The ideal unwrapped signals corresponding to the wrapped signals in Figure 4.1. The top-left signal has no noise, the top-right has low noise, the bottom left has medium noise, and the bottom right has high noise.

4.1.1 Polynomial-Based Region-Growing

The adaptive method of *a priori* variance estimation was much more successful than estimating a constant *a priori* variance factor. As a result, the adaptive *a priori* variance factor estimation is used here.

The PBRGPU algorithm consistently performed better than the RGPU algorithm (refer to Table 4.2 for a summary of testing results). In low noise environments, the PBRGPU algorithm performs similarly to the standard RGPU algorithm, but there is a significant improvement over the RGPU algorithm in high noise environments. The improvement to the RMSE of the solution might be expected to come with a higher proportion of identified high-noise pixels. However, with the simulated data, the PBRGPU algorithm does not exhibit this problem, excluding a similar proportion of pixels to the RGPU algorithm.

This suggests that the patch-based polynomial regression provides a more robust phase prediction than the "prediction lines" approach employed by the RGPU algorithm. This is expected, given that the polynomial regression includes *all* unwrapped phase values in the neighbourhood around

the pixel being unwrapped, and also accounts for non-linear local variations in the unwrapped phase function.

In addition, the definition of the test statistics in the PBRGPU algorithm allows for more rigorous detection of high-noise pixels than in the standard RGPU algorithm. Choosing a statistical significance level for the detection of high-noise pixels can be a bit counter-intuitive, since false negatives have a significant impact on path-based solutions. This requires a very strict statistical test, with a small confidence interval. When running the PBRGPU algorithm, a confidence interval of 50 % was used.

Table 4.2: Comparison of Region-Growing Phase Unwrapping (RGPU) and Polynomial-Based Region-Growing Phase Unwrapping (PBRGPU) results for the simulated data.

Noise Level	Root Mean Square Error (Radians)		Percent of Excluded Pixels		Observable Discontinuities	
	PBRGPU	RGPU	PBRGPU	RGPU	PBRGPU	RGPU
None	0.000	0.000	0.0	0.0	No	No
Low	0.006	0.000	0.3	0.0	No	No
Medium	0.094	0.905	3.8	4.0	No	Yes
High	2.224	9.318	21.7	25.6	Yes	Yes

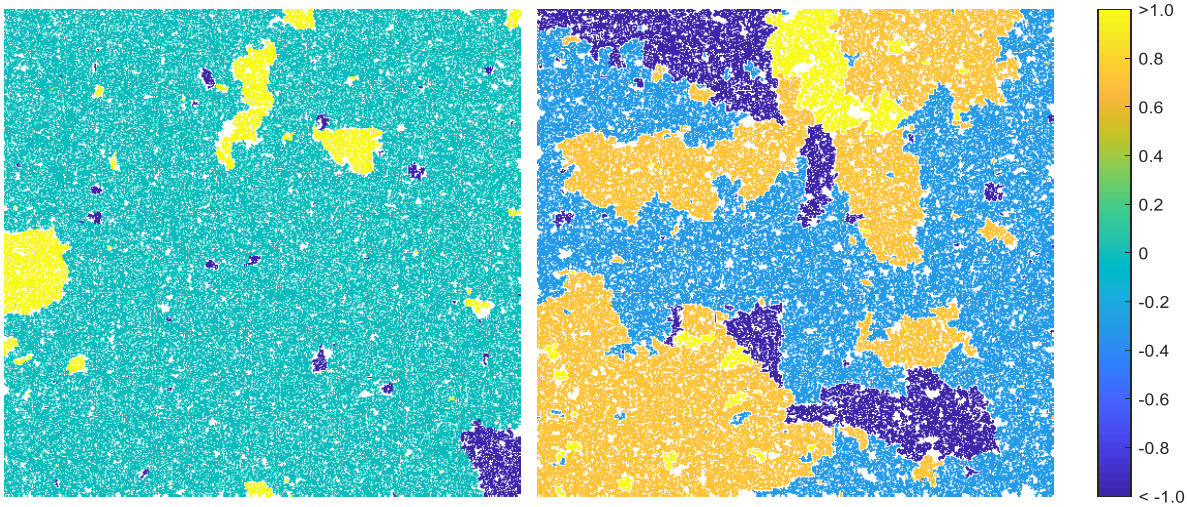

Figure 4.3: The differences between the PBRGPU unwrapped signal and the ideal unwrapped signal (left) and the differences between the RGPU unwrapped signal and the ideal unwrapped signal (right) for the high noise level simulated interferogram.

Even though there are discontinuities introduced by the PBRGPU method under high noise levels, the resulting discontinuous regions are much smaller and more contained than the discontinuous regions produced by the RGPU algorithm. Figure 4.3 highlights the differences between the unwrapped signals and their ideal unwrapped value. Looking at the differences between the PBRGPU unwrapped signal and the ideal signal, it is clear that most of the discontinuous regions form closed areas within the interferogram, and that the size of these regions is smaller than the size of the discontinuous regions produced by the RGPU algorithm.

4.1.2 Path-Based Least-Squares Phase Unwrapping

Since the PBLSPU approach is both a global optimization and a path-based approach, the PBLSPU unwrapping results were compared with those from both the LSPU approach (global optimization) and the RGPU approach (path-based integration).

The PBLSPU approach compares favourably to both the LSPU and RGPU approaches, with a significantly lower RMSE. At a high noise level, the PBLSPU algorithm performs particularly well compared to the other approaches (including the PBRGPU approach). However, it is important to recognize that the simulated wrapped signals were generated using normally-distributed noise, and SAR interferograms typically contain speckle noise.

As with the PBRGPU approach, the introduction of the rigorous detection of high-noise pixels through statistical testing greatly benefits the unwrapping quality. The path-based structure of the PBLSPU approach allows for the complete removal of high-noise pixels, which was not possible under the traditional unweighted/weighted LSPU approaches.

Table 4.3: Comparison of Path-Based Least-Squares Phase Unwrapping (PBLSPU) to standard Least-Squares Phase Unwrapping (LSPU) and Region-Growing Phase Unwrapping (RGPU) results for the simulated data.

Noise Level	**Root Mean Square Error (Radians)**			**Percent of Excluded Pixels**		**Observable Discontinuities**	
	PBLSPU	**LSPU**	**RGPU**	**PBLSPU**	**RGPU**	**PBLSPU**	**RGPU**
None	0.006	0.000	0.000	0.0	0.0	No	No
Low	0.000	0.000	0.006	0.0	0.0	No	No
Medium	0.013	0.264	0.094	1.4	4.0	No	Yes
High	0.559	3.217	2.224	24.9	25.6	Yes	Yes

Given that the proposed least-squares approach is path-based, any individual high-noise pixel that is included in the solution will have a larger impact than it would have for the standard LSPU approach. This issue can manifest itself by producing discontinuous regions when applying the PBLSPU approach, which is a problem shared by any path-based algorithm. While the LSPU solution does not have this issue, (it is constrained to be a smooth surface) the overall effect of outlying pixels is much larger than for the PBLSPU approach. Figure 4.4 clearly shows the improved accuracy of the PBLSPU approach for high noise levels, despite the introduction of discontinuous regions in the unwrapped interferogram.

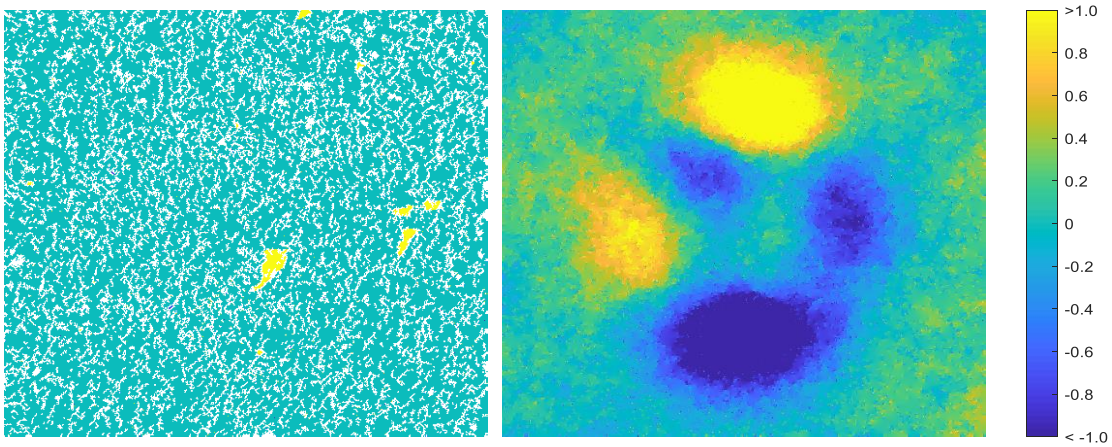

Figure 4.4: The differences between the PBLSPU unwrapped signal and the ideal unwrapped signal (left)and the differences between the LSPU unwrapped signal and the ideal unwrapped signal (right) for the high noise level simulated interferogram.

As with the PBRGPU algorithm, the PBLSPU algorithm requires a small confidence interval for the detection of high-noise pixels due to the prevalence of outlying pixels. When running the PBLSPU algorithm, a confidence interval of 50 % was used.

4.2 RADARSAT-2 Data

The proposed algorithms were also tested using RADARSAT-2 data collected over Polar Bear Provincial Park in Northern Ontario in 2011. The interferograms that were used were all approximately 6000 x 3000 pixels (each pixel represents a 6m x 6m area on Earth's surface), and were produced using PCI's Geomatica software. To determine the algorithms' performance for a variety of signal qualities, three representative SAR image pairs were used for testing: an overall low coherence interferogram (with images collected on June 23, 2011 and September 3, 2011 – average coherence of 38.3 %), an overall high coherence interferogram (with images collected on June 23, 2011 and July 17, 2011 –

average coherence of 41.3 %), and an interferogram with coherence that varies significantly over its extent (with images collected on June 23, 2011 and August 10, 2011 – average coherence of 38.6 %). Figures 4.5, 4.6, and 4.7 show the coherence maps and the wrapped interferometric phases of the three interferograms used to test the proposed algorithms.

Colour scales for the unwrapped signals are provided, but are not directly comparable, since signals are unwrapped relative to the seeded pixels. Despite this, the range of the data defined by the colour scales provides information about whether there were any serious issues in the unwrapping process (an extremely large range indicates that discontinuities were introduced through the unwrapping process).

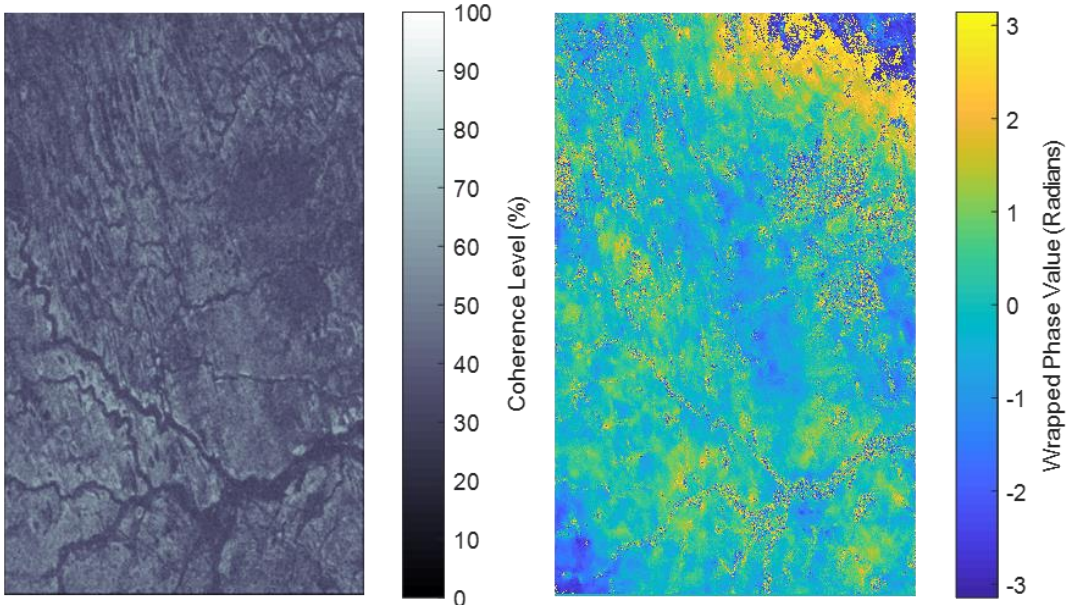

Figure 4.5: High coherence interferogram. The coherence values for the interferogram are shown in the left image, and the wrapped phase values are shown in the right image. Images were collected on June 23 and July 17, 2011.

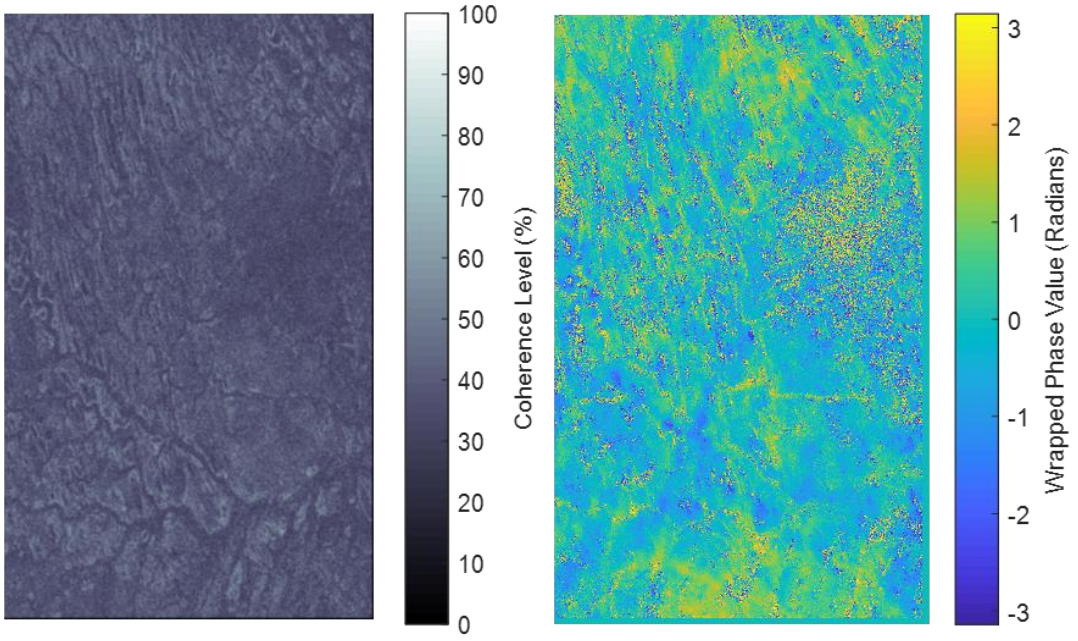

Figure 4.6: Low coherence interferogram. The coherence values for the interferogram are shown in the left image, and the wrapped phase values are shown in the right image. Images were collected on June 23 and September 3, 2011.

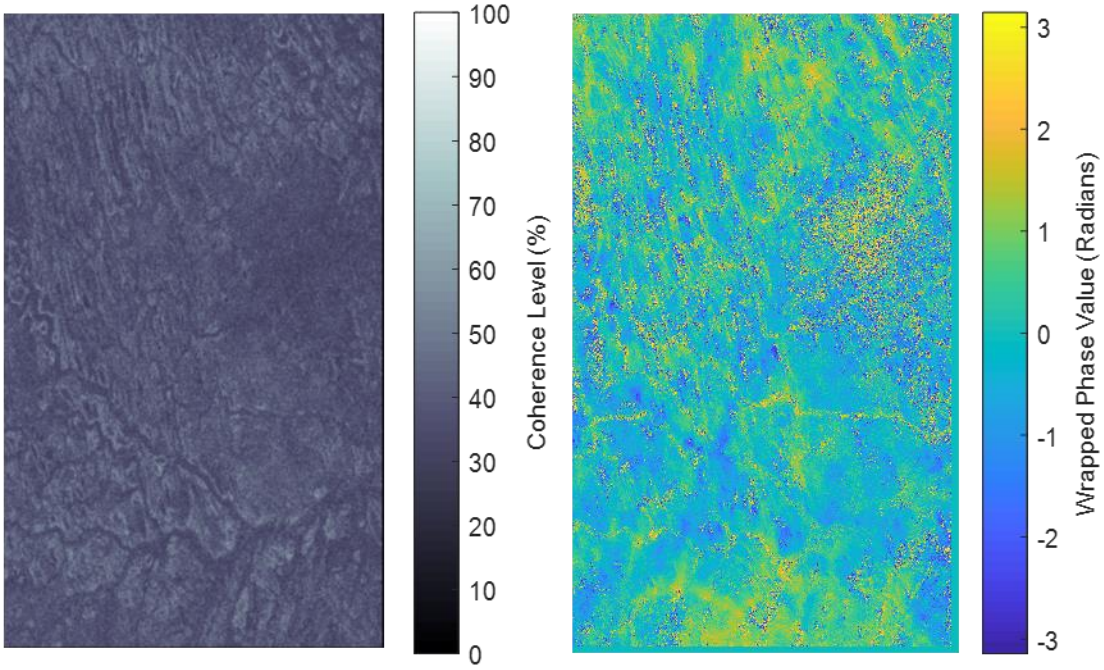

Figure 4.7: Varied coherence interferogram. The coherence values for the interferogram are shown in the left image, and the wrapped phase values are shown in the right image. Images were collected on June 23 and August 10, 2011.

4.2.1 Polynomial-Based Region-Growing Phase Unwrapping

As with the simulated data, the PBRGPU algorithm consistently performs better than the RGPU algorithm for the sample SAR interferograms (see Figures 4.8, 4.9, and 4.10 for a visual comparison). Even with the strictest parameters put forth in Xu's [1999] documentation of the RGPU algorithm, there are many discontinuities introduced.

While the RGPU algorithm excludes far fewer pixels, the excluded pixels appear to form unnatural shapes (e.g. straight horizontal/vertical lines). The excluded pixels for the PBRGPU algorithm, however, appear to align quite well with low coherence regions in the interferogram. This reduces the need for masking low coherence regions prior to the unwrapping process.

Looking at the colour scale for the unwrapped signals in Figure 4.8, 4.9, and 4.10, the RGPU unwrapped signal has a range that is unreasonable when compared to the wrapped signals. The range of the unwrapped signal produced by the PBRGPU approach is much more feasible for this data.

With the high coherence signal, the PBRGPU algorithm did introduce observable discontinuities. It is worth noting, though, that the very small discontinuous region (found in the upper-left corner of the PBRGPU-unwrapped signal in Figure 4.8) is a much less severe issue than the major discontinuities introduced in the RGPU-unwrapped signal.

As with other path-based approaches, if there are any unwrapping errors, then the unwrapped signal is dependent upon the selected seed pixels. Since the relationship between the seed pixels and the unwrapped signal is non-deterministic, it is not possible to determine an optimal set of seed pixels that minimizes the overall unwrapping error. To mitigate this issue, seed pixels are selected at random from a set of candidate pixels that all have a low variance in their neighbourhood.

Table 4.4: Comparison of PBLSPU and RGPU results for the interferograms derived from RADARSAT-2 data for Polar Bear Provincial Park.

Coherence Level	Percent of Excluded Pixels		Observable Discontinuities	
	PBRGPU	**RGPU**	**PBRGPU**	**RGPU**
High	19.1	6.7	Yes	Yes
Low	39.4	9.8	No	Yes
Varied	41.3	12.6	No	Yes

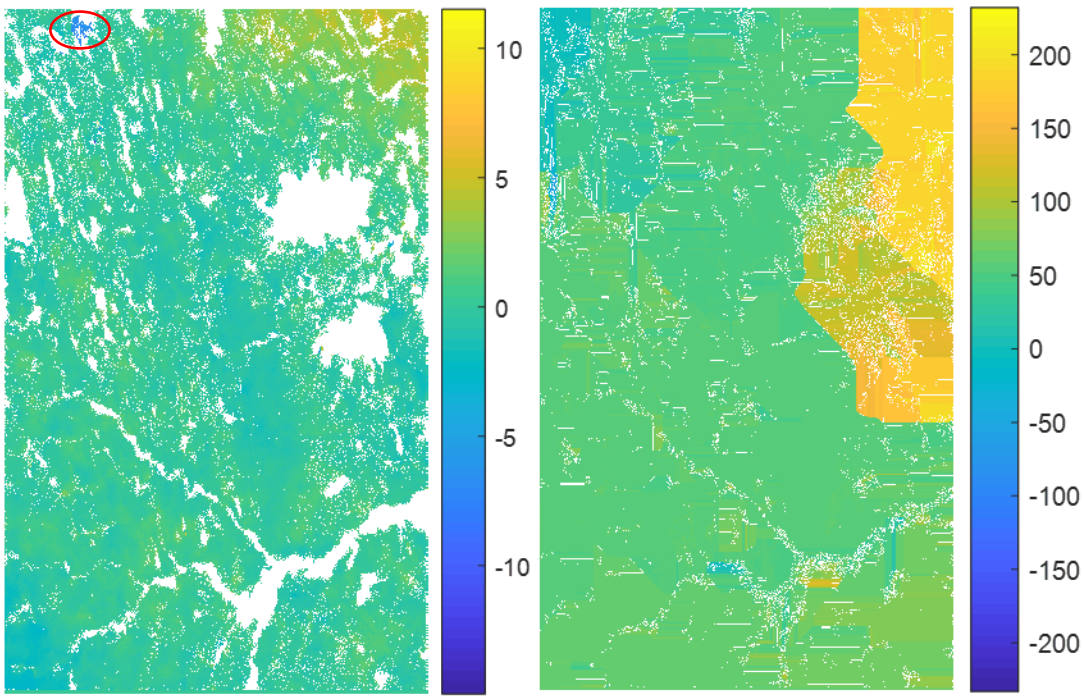

Figure 4.8: Unwrapped phase determined for the high coherence interferogram using PBRGPU (left) and RGPU (right), in radians. Excluded pixels are shown in white. Note that the colourbars are not directly comparable, but provide information about the range of the unwrapped data.

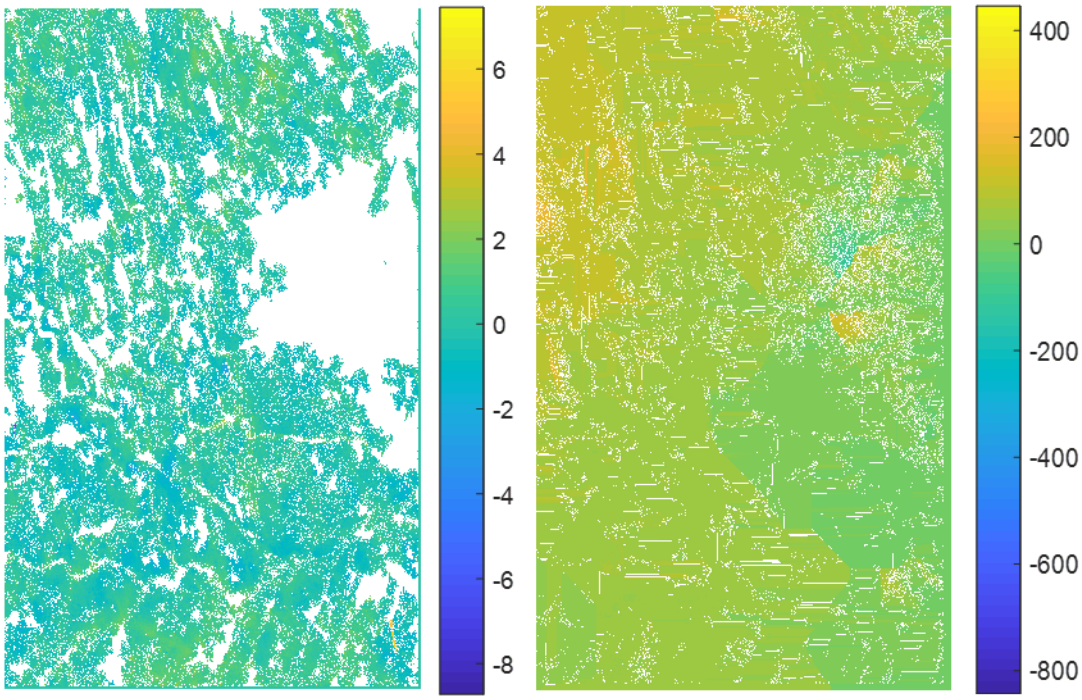

Figure 4.9: Unwrapped phase determined for the low coherence interferogram using PBRGPU (left) and RGPU (right), in radians. Excluded pixels are shown in white. Note that the colourbars are not directly comparable, but provide information about the range of the unwrapped data.

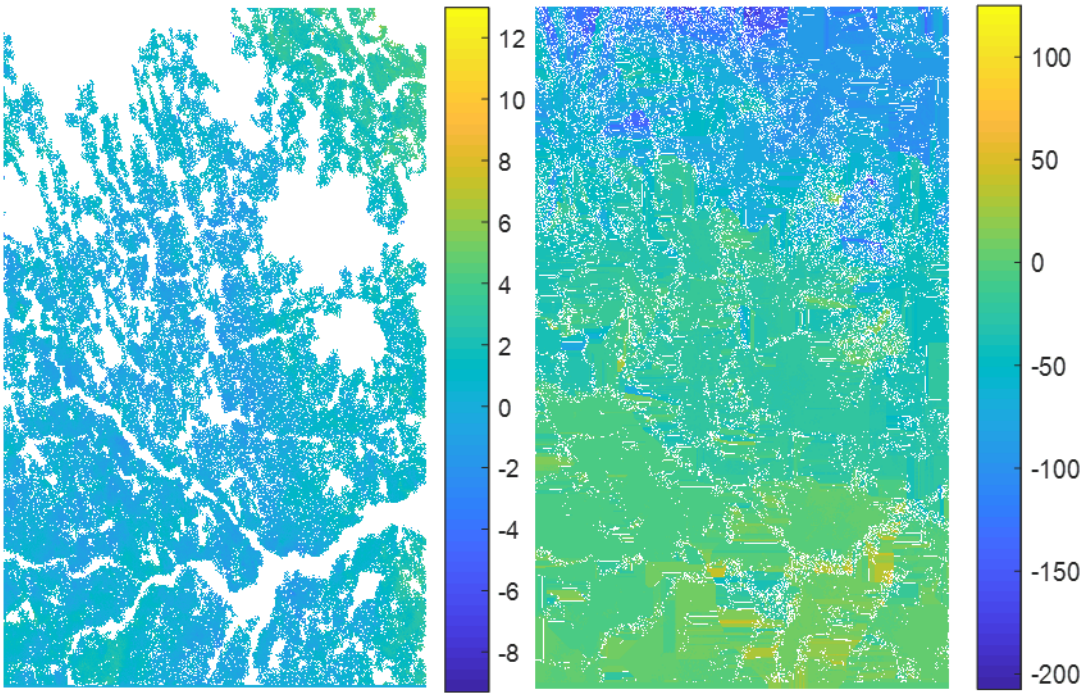

Figure 4.10: Unwrapped phase determined for the varied coherence interferogram using PBRGPU (left) and RGPU (right), in radians. Excluded pixels are shown in white. Note that the colourbars are not directly comparable, but provide information about the range of the unwrapped data.

4.2.2 Path-Based Least-Squares Phase Unwrapping

The PBLSPU algorithm also performs quite well for the sample SAR interferograms. The PBLSPU and PBRGPU unwrapped signals agree very well with one another, although the PBLSPU approach appears to exclude a significantly larger portion of the interferogram in the unwrapping process.

Given that the PBLSPU algorithm is a path-based approach, it can introduce discontinuities in the unwrapping process. This can be seen in the low coherence and varied coherence interferograms, shown in Figures 4.12 and 4.13, respectively. The standard LSPU algorithm, however, will always provide a smooth unwrapped signal, but the prevalence of residue points in the interferogram produces inaccuracies in the standard LSPU unwrapped signal.

One way of mitigating the issue of discontinuous regions in the PBLSPU approach is to use a larger patch size when applying the least-squares matrix equations in order to allow for a higher redundancy in the matrix equations, and in turn to allow for better quality control. This, of course, comes with a longer processing time, but can also increase the likelihood of a Type II error in the detection of high-noise pixels, because the wrapped pixels are revisited whenever they are within the processing patch surrounding a seed pixel. For example, for a 3 x 3 patch size, each pixel is visited a

maximum of 9 times, but for an 11 x 11 patch size, each pixel is visited a maximum of 121 times. For the sample interferograms, PBLSPU performed best when using a 7 x 7 patch.

As an additional test of the unwrapping accuracy, both the PBLSPU and LSPU unwrapped signals were used to simulate the vertical and horizontal phase difference observations used in the least-squares models. The differences between these simulated values and the wrapped phase difference values from the wrapped interferograms were then used to estimate the *a posteriori* standard deviation of the wrapped phase difference values for each method. The *a posteriori* standard deviation determined by for the PBLSPU unwrapped signal ranges from being ten to seventy times smaller than the *a posteriori* standard deviation determined for the LSPU unwrapped signal. This indicates a significant improvement in the quality of the unwrapped signal.

Table 4.5: Comparison of PBLSPU, RGPU, and LSPU results for the interferograms derived from RADARSAT-2 data for Polar Bear Provincial Park

Coherence Level	Percent of Excluded Pixels		Observable Discontinuities		*A Posteriori* Standard Deviation (Radians)	
	PBLSPU	RGPU	PBLSPU	RGPU	PBLSPU	LSPU
High	40.4	6.7	No	Yes	0.019	1.319
Low	39.2	9.8	Yes	Yes	0.031	0.729
Varied	55.2	12.6	Yes	Yes	0.056	0.609

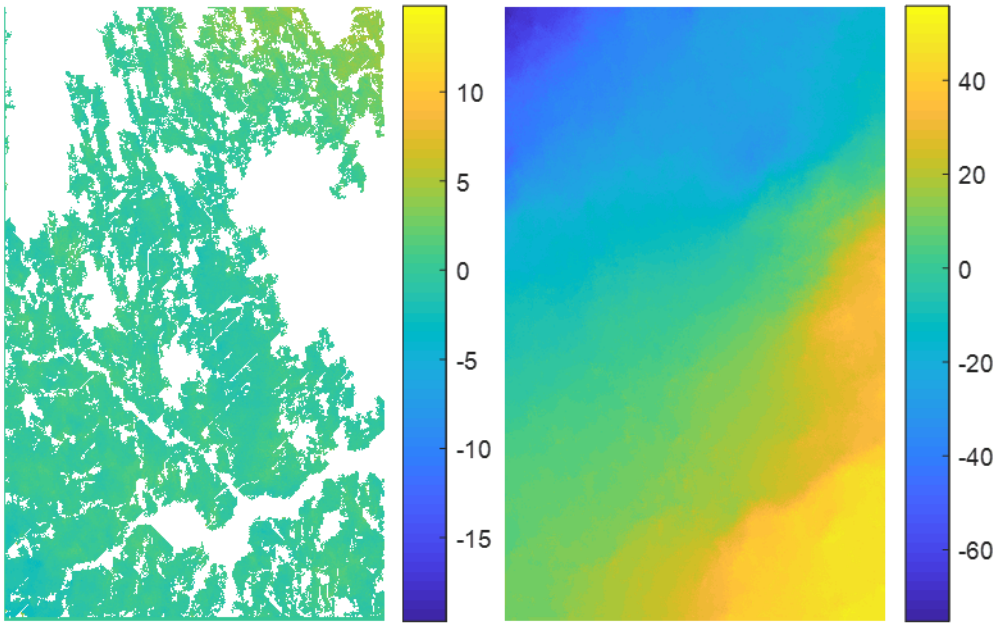

Figure 4.11: Unwrapped phase determined for high coherence interferogram using the PBLSPU (left) and LSPU (right) methods, in radians. Excluded pixels are shown in white. Note that the colourbars are not directly comparable, but provide information about the range of the unwrapped data.

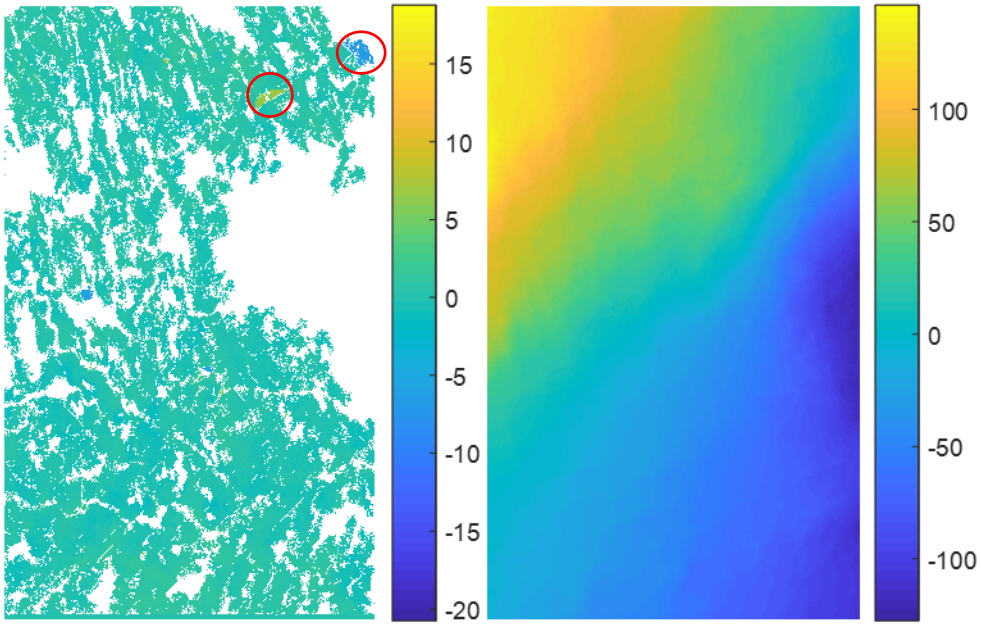

Figure 4.12: Unwrapped phase determined for low coherence interferogram using the PBLSPU (left) and LSPU (right) methods, in radians. Excluded pixels are shown in white. Note that the colourbars are not directly comparable, but provide information about the range of the unwrapped data.

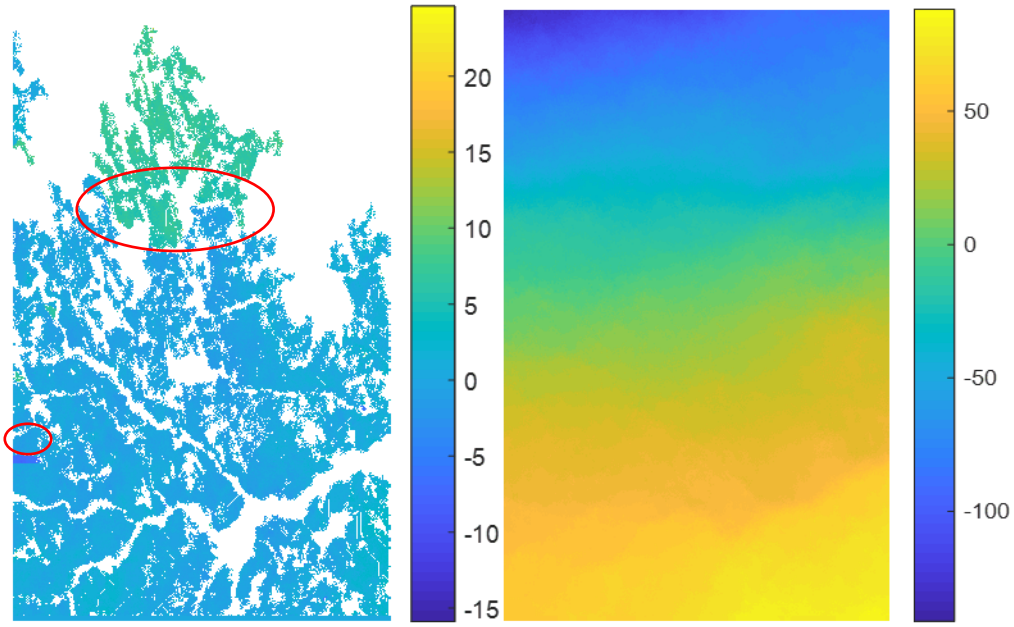

Figure 4.13: Unwrapped phase determined for varied coherence interferogram using the PBLSPU (left) and LSPU (right) methods, in radians. Excluded pixels are shown in white. Note that the colourbars are not directly comparable, but provide information about the range of the unwrapped data.

4.2.3 Comparison to SNAPHU Algorithm

To determine how well the proposed algorithms perform, a comparison with the widely applied SNAPHU algorithm was conducted. The sample interferograms were unwrapped with the SNAPHU algorithm using PCI's Geomatica InSAR processing tools. SNAPHU performed quite well overall, and for the most part produced a smooth unwrapped phase function. The algorithm did struggle in low coherence areas, as was apparent in the rightmost side of the unwrapped interferograms presented in Figures 4.14, 4.15, and 4.16. In these regions, discontinuities were commonly introduced by the SNAPHU algorithm, but tended to be very restricted in size.

A more concerning issue that arises when one is unwrapping the varied coherence interferogram, is that broad discontinuous regions are introduced by the SNAPHU algorithm for this signal. This is likely due to branch cuts being formed such that these broad regions are completely disconnected from one another. The PBLSPU approach also encountered this issue, although to a lesser extent. The PBRGPU algorithm appears to resolve this issue entirely for the sample data and, in the event of multiple seeded regions not being able to be merged, the algorithm is well-suited to identifying disconnected regions and allowing for these regions to be processed separately.

Masking low coherence regions in the interferogram would improve the SNAPHU unwrapping results, but the PBRGPU and PBLSPU algorithms achieve high-accuracy results without masking. Unwrapping without masking allows for the unwrapping process to penetrate through low coherence regions, and ensures that potentially valuable data are not being thrown out.

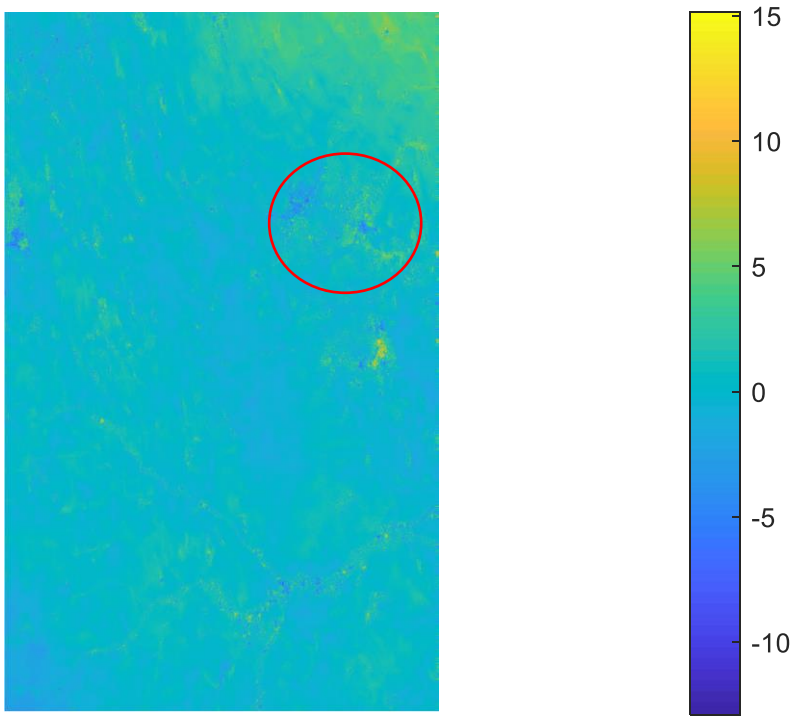

Figure 4.14: SNAPHU unwrapped phase values for the high coherence interferogram, in radians. Note that the colourbars are not directly comparable, but provide information about the range of the unwrapped data.

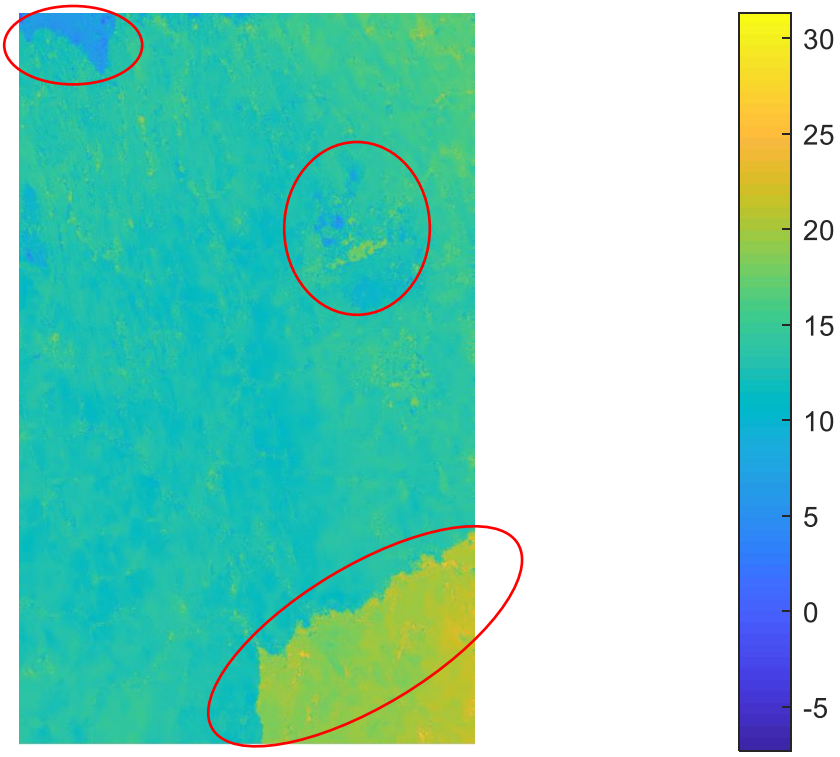

Figure 4.15: SNAPHU unwrapped phase values for the varied coherence interferogram, in radians. Note that the colourbars are not directly comparable, but provide information about the range of the unwrapped data.

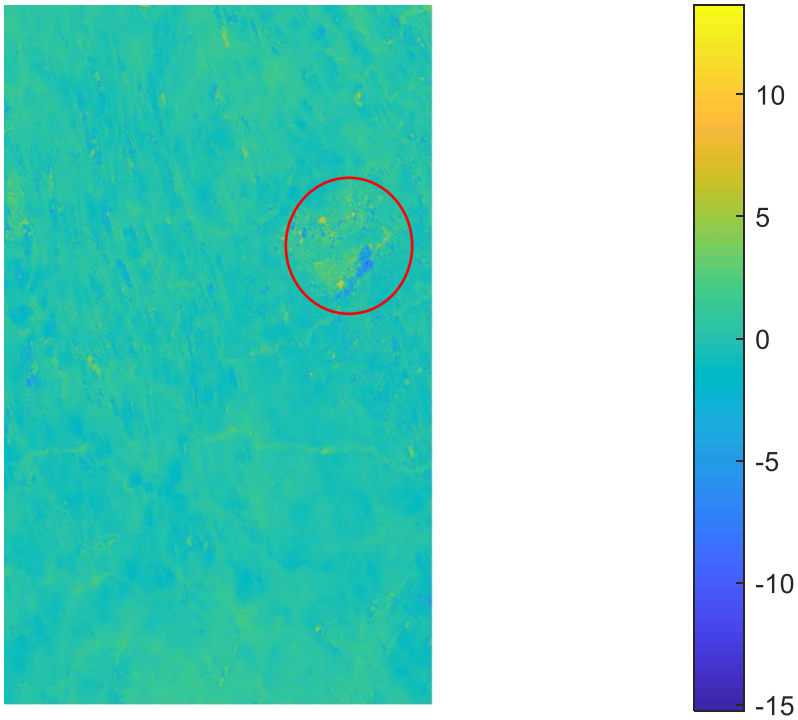

Figure 4.16: SNAPHU unwrapped phase values for the low coherence interferogram, in radians. Note that the colourbars are not directly comparable, but provide information about the range of the unwrapped data.

4.2.4 Comparison Between Proposed Approaches

Both of the proposed phase unwrapping approaches share many of the same strengths, since they were constructed using the same set of guiding principles. Both approaches improve upon the detection and removal of high-noise pixels, to the point that neither algorithm requires low-coherence thresholding, even for low-quality interferograms.

The biggest differences between the two approaches lie in the domain of the unwrapped signal and in the prevalence of discontinuities in the unwrapped signal.

Both approaches remove high-noise pixels that have been identified as the phase unwrapping process progresses. The PBLSPU approach excludes broader regions of pixels from the unwrapping solution than the PBRGPU approach does (compare Figure 4.8 with Figure 4.11, Figure 4.9 with Figure 4.12, and Figure 4.10 with Figure 4.13), which suggests that the PBRGPU approach penetrates into low-coherence regions better than the PBLSPU approach does. Another notable difference in the domain of the unwrapped signal is that the PBLSPU approach has fewer "gaps" in the unwrapped signal than the PBRGPU approach. Looking at the PBRGPU unwrapped signals (Figures 4.8 to 4.10), a speckling pattern of excluded pixels is noticeable (in Figures 4.8 to 4.10, this is shown as a white speckling pattern). This

pattern is mostly absent from the PBLSPU unwrapped signals (Figures 4.11 to 4.13), which indicates that the PBLSPU approach handles isolated high-noise pixels better than the PBRGPU approach does.

The PBLSPU approach tends to introduce more discontinuities than the PBRGPU approach does. This makes the PBLSPU approach less reliable for unwrapping SAR interferograms than the PBRGPU approach in its current state. It is worth noting, however, that for the normally-distributed simulated wrapped signals, the PBLSPU unwrapped signal has a much higher accuracy than the PBRGPU unwrapped signal (comparing the root mean square errors in Tables 4.2 and 4.3).

4.2.5 On Local Normality of the Interferometric Phase

Both proposed algorithms rely on the interferometric phase following a normal distribution in the processing patch surrounding each seed pixel. The parameters defining these normal distributions can change throughout the interferogram, but this condition must be satisfied locally for the quality assurance used in the phase unwrapping process to be statistically rigorous.

To test this assumption for the PBRGPU algorithm, differences between the unwrapped phase values and their predicted values were recorded for every iteration of the algorithm. These values were standardized by their standard deviations in order to create the histogram shown in Figure 4.17. Note that these values should follow a *t*-distribution, since they are standardized by their sample standard deviations. Note that none of these standardized differences exceed $\pm$ 0.5. This is due to the fact that any values outside of this range were considered high-noise pixels in the phase unwrapping process.

Assuming that the interferometric phase in the neighbourhood of each seed pixel follows a normal distribution, these normalized differences should each follow a t-distribution with degrees of freedom equal to the degrees of freedom of the polynomial fitting.

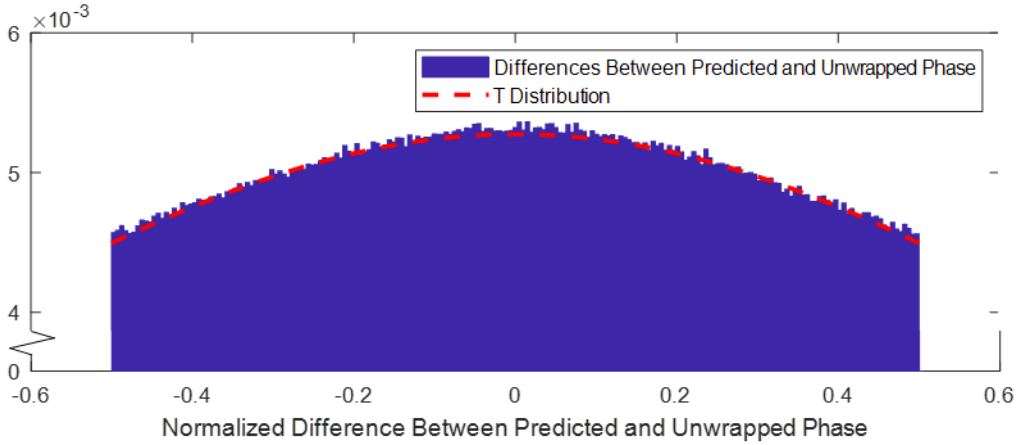

Figure 4.17: Comparison between the histogram of differences between predicted and unwrapped phase values for the high coherency interferogram and a t-distribution with 3 degrees of freedom for the PBRGPU algorithm.

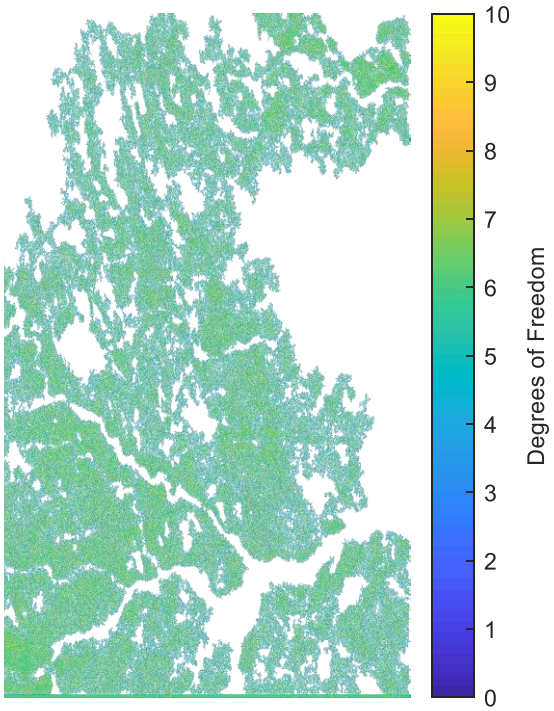

Figure 4.18: The degrees of freedom of the polynomial fitting process for interferogram patches in the high coherency interferogram.

The degrees of freedom mostly varied from one to five, with the average degrees of freedom of the polynomial fitting process being three (see Figure 4.18). To test the assumption of normality of the data, a Chi-square goodness of fit test (Pearson distribution test) was conducted against the theoretical distribution of the data defined by a *t*-distribution with three degrees of freedom. From Figure 4.17, it is clear that the posited distribution (shown as a dotted red line) follows the data (shown as the bars of the histogram) closely, but it was found to statistically deviate from the histogram of the data.

From this, it can be concluded that the inlying interferometric phases for each interferogram patch do not follow a normal distribution. The quality assurance measures used in the proposed algorithms assume that the interferometric phase follows a normal distribution, and this assumption falls short in describing the probability distribution of the SAR data. It is important to note that while a normal distribution model is insufficient for describing the SAR data, it serves as a reasonable approximation to its statistical distribution.

5 Conclusion and Remarks

Both of the proposed algorithms performed well for both the simulated data and the real data from Polar Bear Provincial Park, providing accuracy improvements in the unwrapped signal while they function as efficiently as the algorithms they are based on (the PBLSPU algorithm takes significantly longer to run than the LSPU algorithm, but runs almost as quickly as the region-growing approaches).

The accuracy of the phase unwrapping solutions from the proposed methods appears to be much higher for the simulated data than for the real SAR data. This can be attributed to the fact that both proposed algorithms were developed to accommodate high levels of normally-distributed noise, and the SAR data contains speckle noise. Although this presents a drawback for both algorithms and requires the use of very low confidence intervals when performing statistical tests, the algorithms still present an improvement over the existing solutions.

The proposed phase unwrapping solutions also seemed to have higher accuracies for the high- and low-coherence interferograms than for the varied coherence interferogram. Since both proposed phase unwrapping algorithms adapt to local noise levels in the interferogram, this suggests that more drastic variations in the coherence values interferes with the algorithms' abilities to accurately identify local noise levels in the interferogram.

It is worth noting that the accuracy of the proposed algorithms is comparable to that of the SNAPHU algorithm, and even surpasses the accuracy of the SNAPHU algorithm in some cases (refer to Section 4.2.3). None of the proposed algorithms requires masking low coherence regions in the interferogram, in contrast to the SNAPHU algorithm, which produces unreliable results in low coherence regions if they are not masked prior to unwrapping.

Both of the proposed algorithms have all thresholds for the detection of high-noise pixels defined in terms of statistical significance levels, which no other phase unwrapping algorithm generally allows for. Ultimately, these rigorously-defined statistical tests make the selection of these thresholds much more intuitive for an end user and definitely improve the detection of high-noise pixels.

One unique contribution of the PBLSPU approach is that it allows for the complete removal of outlying pixels in a least-squares model, which was not possible using the standard least-squares phase unwrapping approaches presented in Section 2.4. This fully addresses the effects of residue points on the least-squares solution, and effectively restricts the impact of residue points to very small regions surrounding the residues.

Choosing a patch size to use for the PBLSPU algorithm is challenging due to the competing demands of wanting higher redundancy in the least-squares equations for more reliable detection/removal of high-noise pixels but also wanting to reduce the number of times each pixel is visited by the algorithm to reduce the chance of a Type II error. A patch size of 7 x 7 pixels appears to be a reasonable compromise between these demands, but there is unfortunately no way to avoid the ideal patch size being an *ad hoc* parameter.

Unfortunately, the statistical testing employed by the proposed methods assumes the SAR data to follow a normal distribution in each interferogram patch. This assumption is incorrect, although it provides a reasonable approximation to the distribution of the data. This and the low degrees of freedom for the phase unwrapping process in any given interferogram patch are the biggest obstacles to the success of the proposed algorithms.

The methods developed through this research could be generalized for multiple-polarization interferograms, which would allow for the incorporation of multiple interferograms to produce a single unwrapped signal. This would increase the degrees of freedom of phase prediction, which would in turn improve the reliability of the process of detecting and removing high-noise pixels. This development would address the main weaknesses of both proposed phase unwrapping approaches and make them more robust to high-noise levels in the wrapped interferograms.

The PBLSPU method could be implemented using a weighted least-squares model, and could make use of variance component estimation to separately define *a posteriori* variance factors for the estimated unwrapped phases and the wrapped phase differences. Both of these modifications to the PBLSPU method would provide more in-depth information regarding the quality of the adjustment process, thereby allowing the detection of high-noise pixels to be a more informed process.

The results of this research show a significant improvement in unwrapping quality over other phase unwrapping approaches, and introduces rigorous statistical testing into the phase unwrapping process. The restructured least-squares approach, in particular, resolves many of the limitations of the standard least-squares phase unwrapping approach. The proposed phase unwrapping algorithms produce accurate unwrapping results, and the rationale that informed these algorithms' development could be applied to improve/develop new phase unwrapping approaches.

References

Alexakis, Dimitrios D, et al. “Soil Moisture Content Estimation Based on Sentinel-1 and Auxiliary Earth Observation Products. A Hydrological Approach.” *Sensors*, vol. 17, no. 6, 2016, doi:10.3390/s17061455.

August, Isaac, Dan Blumberg and Stanley Rotman. Subpixel high accuracy image registration for radar interferometry processes. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. 38. 2010

Bähr, Hermann, and Ramon F. Hanssen. “Reliable Estimation of Orbit Errors in Spaceborne SAR Interferometry.” *Journal of Geodesy*, vol. 86, no. 12, 2012, pp. 1147–1164., doi:10.1007/s00190012-0571-6.

Bektas, Sebahattin and Yasemin Sisman, The comparison of L1 and L2-norm minimization methods, October 2010, International journal of physical sciences 5(11):1721-1727.

Čapková, Ivana. “Satellite Orbit Errors and Their Influence on Interferograms.” *Department of Mapping and Cartography, Czech Technical University*, 2005.

Cazals, Cécile, et al. “Mapping and Characterization of Hydrological Dynamics in a Coastal Marsh Using High Temporal Resolution Sentinel-1A Images.” *Remote Sensing*, vol. 8, no. 7, 2016, doi:10.3390/rs8070570.

Chen, Curtis W., and Howard A. Zebker. “Network Approaches to Two-Dimensional Phase Unwrapping: Intractability and Two New Algorithms.” *Journal of the Optical Society of America* A, vol. 17, no. 3, 2000, p. 401., doi:10.1364/josaa.17.000401.

Ferretti, Alessandro, et al. “InSAR Principles: Guidelines for SAR Interferometry Processing and Interpretation.” *European Space Agency*, 2007.

Flynn, Thomas J. “Two-Dimensional Phase Unwrapping with Minimum Weighted Discontinuity.” *Journal of the Optical Society of America A*, vol. 14, no. 10, 10 Oct. 1997, p. 2692., doi:10.1364/josaa.14.002692.

Gao, J, et al. “Two-Dimensional Phase Unwrapping Method Using Cost Function of L0 Norm.” *IOP Conference Series: Earth and Environmental Science*, vol. 57, 2017, p. 012041., doi:10.1088/1755-1315/57/1/012041.

Ghiglia, Dennis C., 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*, vol. 11, no. 1, Jan. 1994, p. 107., doi:10.1364/josaa.11.000107.

Ghiglia, Dennis C., and Louis A. Romero. “Minimum Lp-norm two-dimensional phase unwrapping.” *Journal of the Optical Society of America A,* vol. 13, no. 10, Oct. 1996, p. 1999.

Gonzalez, Adriana, and Laurent Jacques. “Robust Phase Unwrapping by Convex Optimization.” *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, doi:10.1109/icip.2014.7025343.

Gutmann, Bernd, and Herbert Weber. “Phase Unwrapping with the Branch-Cut Method: Clustering of Discontinuity Sources and Reverse Simulated Annealing.” *Applied Optics*, vol. 38, no. 26, 10 Sept. 1999, p. 5577., doi:10.1364/ao.38.005577.

Hunt, B. R. “Matrix Formulation of the Reconstruction of Phase Values from Phase Differences.” *Journal of the Optical Society of America*, vol. 69, no. 3, Mar. 1979, p. 393., doi:10.1364/josa.69.000393.

Huntley, J. M. “Noise-Immune Phase Unwrapping Algorithm.” *Applied Optics*, vol. 28, no. 15, 15 Aug. 1989, pp. 3268–3270., doi:10.1364/AO.28.003268.

Karout, Salah A., et al. “Residue Vector, an Approach to Branch-Cut Placement in Phase Unwrapping: Theoretical Study.” *Applied Optics*, vol. 46, no. 21, 20 July 2007, p. 4712., doi:10.1364/ao.46.004712.

Li, J, and S Wang. “Mapping Water Bodies Using SAR Imagery - an Application over the Spiritwood Valley Aquifer, Manitoba.” *Geomatics Canada*, 2017, doi:10.4095/300213.

Li, Qingliang, et al. “A New Fast Quality-Guided Flood-Fill Phase Unwrapping Algorithm.” *Journal of Physics: Conference Series*, vol. 1069, 2018, p. 012182., doi:10.1088/1742-6596/1069/1/012182.

Li, Zhengxiao and James Bethel. Image coregistration in SAR interferometry. Proc. Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.. 2008

Lin, Qian, et al. “Phase Unwrapping through Fringe-Line Detection in Synthetic Aperture Radar Interferometry.” *Applied Optics*, vol. 33, no. 2, 10 Jan. 1994, pp. 201–208., doi:10.1364/ao.33.000201.

Pepe, Antonio, and Fabiana Calo. “A Review of Interferometric Synthetic Aperture RADAR (InSAR) Multi Track Approaches for the Retrieval of Earth’s Surface Displacements.” *Applied Sciences*, vol. 7, no. 12, 2017, doi:10.3390/app7121264.

Sun, Luyi, and Jan-Peter Muller. “Evaluation of the Use of Sub-Pixel Offset Tracking Techniques to Monitor Landslides in Densely Vegetated Steeply Sloped Areas.” *Remote Sensing*, vol. 8, no. 8, 2016, doi:10.3390/rs8080659.

Tian, Xin, et al. “Modeling Orbital Error in InSAR Interferogram Using Frequency and Spatial Domain Based Methods.” *Remote Sensing*, vol. 10, no. 4, 2018, doi:10.3390/rs10040508.

Wessels, John. “InSAR User Manual Geomatica 2017.” *PCI Geomatica*, Apr. 2017.

Winsvold, Solveig H., et al. “Using SAR Satellite Data Time-Series for Regional Glacier Mapping.” *The Cryosphere Discussions*, 2017, doi:10.5194/tc-2017-136.

Xie, Xianming. “Iterated Unscented Kalman Filter for Phase Unwrapping of Interferometric Fringes.” *Optics Express*, vol. 24, no. 17, 2016, p. 18872., doi:10.1364/oe.24.018872

Xu, Wei, and I. Cumming. “A Region Growing Algorithm for InSAR Phase Unwrapping.” *IGARSS 96. 1996 International Geoscience and Remote Sensing Symposium*, doi:10.1109/igarss.1996.516883.

Ying, Leslie. “Phase Unwrapping.” *Wiley Encyclopedia of Biomedical Engineering*, Apr. 2006, doi:10.1002/9780471740360.ebs1356.

Zheng, Dongliang, and Feipeng Da. “A Novel Algorithm for Branch Cut Phase Unwrapping.” *Optics and Lasers in Engineering*, vol. 49, no. 5, 2011, pp. 609–617., doi:10.1016/j.optlaseng.2011.01.017.