INVESTIGATION OF POLARIMETRIC COHERENCE OPTIMIZATION IN PERSISTENT SCATTERER INTERFEROMETRY

A Thesis presented to the Faculty of the Graduate School
University of Missouri-Columbia

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

---

by
JACOB A. GARNER
Dr. Justin J. Legarsky, Thesis Supervisor
DECEMBER 2010

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

INVESTIGATION OF POLARIMETRIC COHERENCE OPTIMIZATION IN PERSISTENT SCATTERER INTERFEROMETRY

presented by Jacob A. Garner, a candidate for the degree of Master of Science, Electrical Engineering, and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Justin Legarsky

Dr. Dominic Ho

Dr. J. Erik Loehr

ACKNOWLEDGMENTS

I would like to express my gratitude and appreciation to several people who helped get me to where I am today. First, I would like to thank my graduate advisor, Dr. Justin Legarsky, for extending his time, patience, and direction to me, as well as for the opportunities that he has presented to me. I would like to thank Dr. J. Erik Loehr for his guidance and encouragement throughout the course of my research. I would like to thank Dr. Dominic Ho for his advice, guidance, and mentoring throughout the course of both my graduate and undergraduate studies. I would also like to thank Dr. Francisco 'Paco' Gomez for always taking the time out his day to share his expert knowledge. Lastly, I wouldn't be where I am today without the love, support, encouragement of my parents, Mark and Beth Garner.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Interferometric Synthetic Aperture Radar (InSAR) provides a means of imaging small deformations of the Earth's surface (millimeter scale). Furthermore, interferometric point target analysis (IPTA) and time-series analyses can be enhanced using polarimetric and interferometric synthetic aperture radar (PolInSAR) processing techniques to improve detection of point targets or coherent scatterers (CS) that are characterized by a point-like scattering behavior. A mixture of single or multiple scattering mechanisms with a high range of backscattering amplitudes characterize many SAR images over a number of areas. However, the fact that a number of scatterers are characterized by a strongly polarized behavior and are located at different heights even within the same resolution cell make the combination of polarimetric and interferometric information a promising one.

For the study site of Socorro, New Mexico, full polarization techniques provided a significant increase (i.e. more than doubled) in CSs compared to a non full polarization. Scattering mechanism and temporal behavior of the CSs were investigated. The CS scattering mechanism behavior was found consistent amongst the CS lists. Full polarization processing was found to increase the CS quantity significantly compared to single polarizations while maintaining high temporal coherence. Thus, this significant increase in full polarization CSs has the potential to increase the overall InSAR processing quality. For sites that challenge (e.g. vegetated areas and other low coherence environments) the single polarization InSAR processing, the full polarization increase in CSs may be essential for processing.

Chapter 1. Introduction

1.1. Overview

Interferometric Synthetic Aperture Radar (InSAR) provides a means of imaging small deformations of the Earth's surface (millimeter scale). Furthermore, interferometric point target analysis (IPTA) and time-series analyses can be enhanced using polarimetric and interferometric synthetic aperture radar (PolInSAR) processing techniques to improve detection of point targets or coherent scatterers (CS) that are characterized by a point-like scattering behavior. A mixture of single or multiple scattering mechanisms with a high range of backscattering amplitudes characterize many SAR images over a number of areas. However, the fact that a number of scatterers are characterized by a strongly polarized behavior and are located at different heights even within the same resolution cell make the combination of polarimetric and interferometric information a promising one.

For enhancing InSAR techniques, this study explores strategies and approaches for detecting CS candidates using InSAR IPTA of fully polarimetric SAR data. IPTA analysis is more accurate and efficient with a more confident list of coherent scatterer candidates. Fully polarimetric data permit a more direct assessment of the scattering properties. Hence, we may assess the enhancement in coherent scatterer point identification using polarimetric criteria. From Radarsat-2 data that we acquired, a Socorro, New Mexico site was selected with fully polarimetric data for five dates.

1.2. EM Wave Propagation and Polarization

Electromagnetic radiation is a phenomenon that transfers energy over a distance, without the transfer of matter. This energy is transferred through self-propagating waves. Remote sensing relies on the use of electromagnetic waves to provide information about physically distant targets to the sensor system. The precise handling and interpretation of these waves is necessary for the sensors to derive accurate measurements from them.

Electromagnetic waves are generated as the result of the interaction between oscillating electric and magnetic fields. The time-harmonic, coupled fields allows for them to propagate as waves. The propagation of electromagnetic waves is a vector phenomenon and can be expressed using complex vectors. Using complex vectors allows both spatial and temporal nature of the waves to be represented. The behavior and interaction of electromagnetic fields is governed by Maxwell's Equations. These equations express the relationship between the electric and magnetic fields as well as their interaction with matter. Maxwell's equations, in differential form, are

$$\nabla \times \vec{E}(\vec{r},t) = -\frac{\partial \vec{B}(\vec{r},t)}{\partial t} \tag{1-1}$$

$$\nabla \times \vec{H}(\vec{r},t) = \vec{J}_T(\vec{r},t) + \frac{\partial \vec{D}(\vec{r},t)}{\partial t} \tag{1-2}$$

$$\nabla \cdot \vec{D}(\vec{r},t) = \rho(\vec{r},t) \tag{1-3}$$

$$\nabla \cdot \vec{B}(\vec{r},t) = 0 \tag{1-4}$$

For the case of time-harmonic electromagnetic fields, Maxwell's equations can be combined to form the wave equation,

$$\Delta\vec{E}(\vec{r},t) - \mu\varepsilon\frac{\partial^2\vec{E}(\vec{r},t)}{\partial t^2} - \mu\sigma\frac{\partial\vec{E}(\vec{r},t)}{\partial t} = -\frac{1}{\varepsilon}\frac{\partial\nabla\rho(\vec{r},t)}{\partial t} \tag{1-5}$$

One particular field configuration that satisfies the wave equation is the monochromatic plane wave solution. The monochromatic plane wave solution is considered in remote sensing applications and polarization analysis due its characteristic of parallel equiphase planes, or wavefronts. For satellite remote sensing applications, plane wave assumption is appropriate due to the large distance between the sensor and the target. In the far-field region of the antenna, the curvature of the radiated wavefronts is no longer apparent for the portion being considered. Thus, the incident waves at the target can be considered as plane waves. As plane waves, the field is no longer a function of the coordinates used to describe each equiphas plane [Balanis, 1989]. The same assumption can be applied for the scattered waves returning to the sensor.

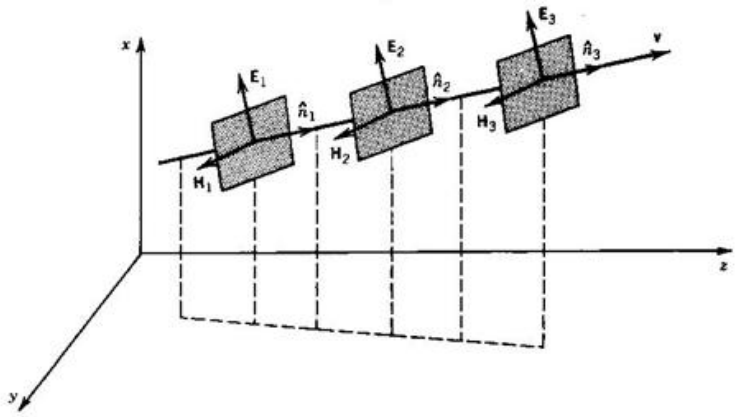


Figure 1. Parallel equiphase planes of an electromagnetic plane wave [Balanis, 1989]

As electromagnetic plane waves, the incident and scattered fields can be expressed as 2D complex vectors. Since both the incident waves at the target, $\vec{E}_I$, and the scattered waves at the sensor, $\vec{E}_S$, can be considered in terms of 2D vectors, the scattering phenomena can be expressed as the transformation of one 2D vector into another 2D vector [Elachi and van Zyl, 2006]. Since the magnetic and electric fields are coupled, they can be described by the electric field vector.

$$\vec{E}_S = [S]\vec{E}_I \tag{1-6}$$

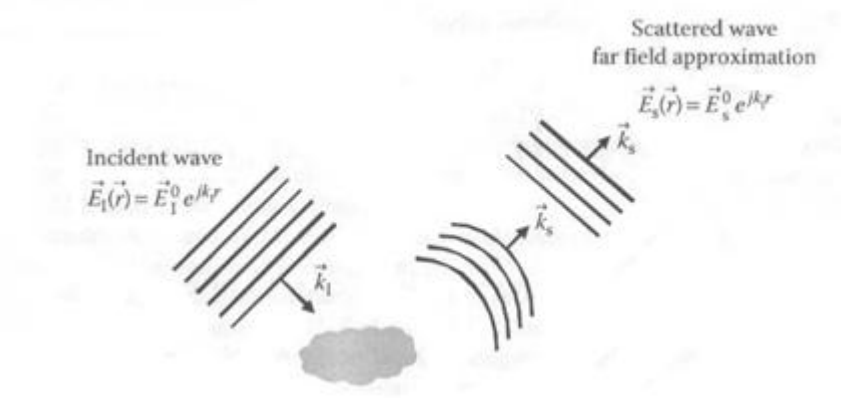


Figure 2. Scattering of incident plane waves [Lee and Pottier, 2009].

Along with amplitude and phase, radiated electromagnetic waves are also characterized by their polarization. Polarization of an electromagnetic wave describes the time-varying direction and relative amplitude of the electric field vector. It is expressed as the locus that the tip of the instantaneous electric field vector traces in time, as observed along the direction of propagation.

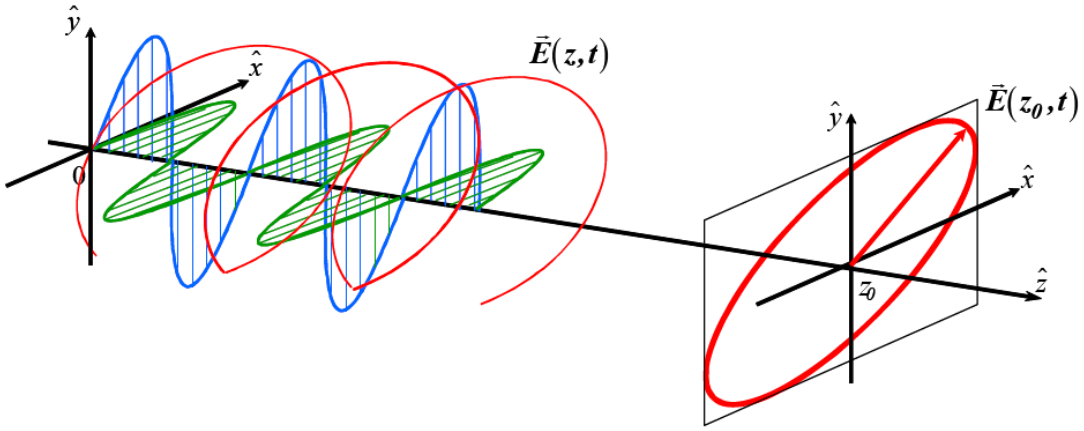


Figure 3. Electric field trajectory of a propagating wave [Lee and Pottier, 2009].

As illustrated in the Figure 3, the tip of the electric field vector traces out a helical trajectory for a propagating wave. By viewing the shape of this helix along the direction of propagation, a 2D representation referred to as the polarization ellipse is formed. This ellipse, along with its ellipicity and orientation, is expressed in terms of the electric field components below.

$$\left[\frac{E_x(z_0,t)}{E_{0x}}\right]^2 - 2\frac{E_x(z_0,t)E_y(z_0,t)}{E_{0x}E_{0y}}\cos(\delta_y-\delta_x) + \left[\frac{E_y(z_0,t)}{E_{0y}}\right]^2 = \sin(\delta_y-\delta_x) \tag{1-7}$$

$$\tan 2\phi = 2\frac{E_{0x}E_{0y}}{E_{0x}^2 - E_{0y}^2}\cos(\delta_y-\delta_x) \tag{1-8}$$

$$\left|\sin 2\tau\right| = 2\frac{E_{0x}E_{0y}}{E_{0x}^2 + E_{0y}^2}\left|\sin(\delta_y-\delta_x)\right| \tag{1-9}$$

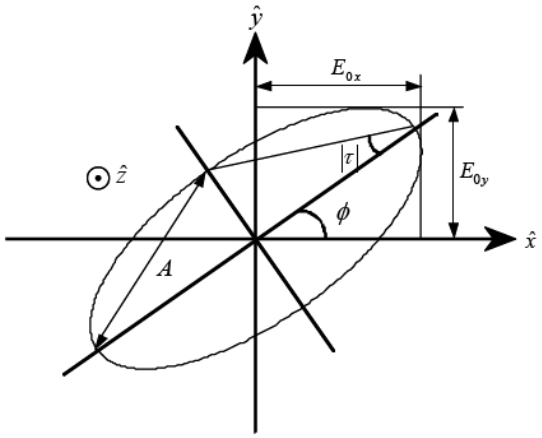


Figure 4. Polarization ellipse [Lee and Pottier, 2009].

When viewed from this perspective, the polarization of a wave falls into one of three classifications: linear, circular, or elliptical. It should be observed that the linear and circular cases are special cases of the more general elliptical case. In addition to shape, polarization is also classified by the direction that the electric field vector rotates. When viewing the polarization ellipse it is necessary to designate the sense of rotation. Clockwise rotation is referred to as a right-hand polarization, and counterclockwise rotation as left-hand polarization.

1.3. Radar Imaging and Synthetic Aperture Radar

Radar imaging is an active form of remote sensing that utilizes radiation in the microwave spectrum to produce images of the target area. As an active sensor system, radar provides its own illumination which gives it the capability to perform regardless of the sun's relative position. Also, by taking advantage of the atmospheric opacity windows in the microwave spectrum, radar is able to operate in wide variety of weather conditions

with very little attenuation [Richards, 2005]. These capabilities give airborne and spaceborne radar imaging systems several advantages over optical imaging systems.

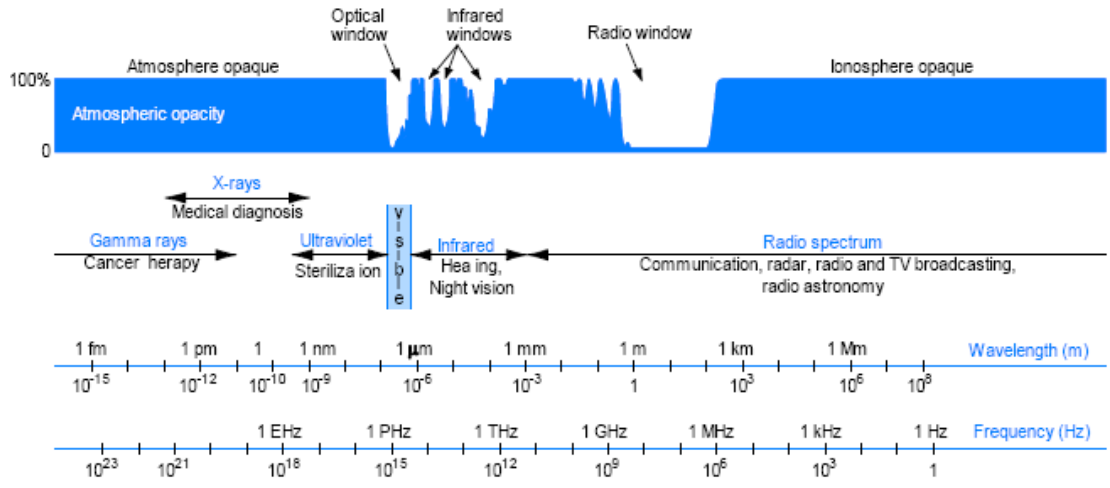


Figure 5. The electromagnetic spectrum and corresponding atmospheric opacity [Ulaby, 2004].

Table 1. Radar frequency band designations [Richards, 2005].

Band	*Frequencies (f)*	*Wavelengths (λ)*
HF	3–30 MHz	100–10 m
VHF	30–300 MHz	10–1 m
UHF	300 MHz–1 GHz	1–30 cm
L	1–2 GHz	30–15 cm
S	2–4 GHz	15–7.5 cm
C	4–8 GHz	7.5–3.75 cm
X	8–12 GHz	3.75–2.5 cm
Ku	12–18 GHz	2.5–1.67 cm
K	18–27 GHz	1.67–1.11 cm
Ka	27–40 GHz	1.11 cm–7.5 mm
mm	40–300 GHz	7.5–1 mm

Traditionally, radar was developed for the purpose of detecting and tracking targets. By systematically scanning and recording over an area, a 2D radar image can generated. For conventional radar imaging, the resolution in the range direction is limited only by the bandwidth of the sensor. However the cross-range, or azimuth, resolution is dependent on the range of the sensor from the target, and degrades as this distance increases. Thus, the azimuth resolution of real-aperture radar systems makes imaging impractical for most airborne and spaceborne applications [Richards, 2005].

To overcome this limitation in azimuth resolution, a technique called Synthetic Aperture Radar (SAR) was developed. Synthetic aperture radar uses the motion of the sensor relative to the target area to artifitially increase the aperture used to image the area. By operating continuously over the length of the synthetic aperture, data acquired at several positions can be combined by the signal processor to create a high resolution image. Effectively, the single antenna of the sensor is used to create a large phased array antenna in order to achieve the desired azimuth resolution.

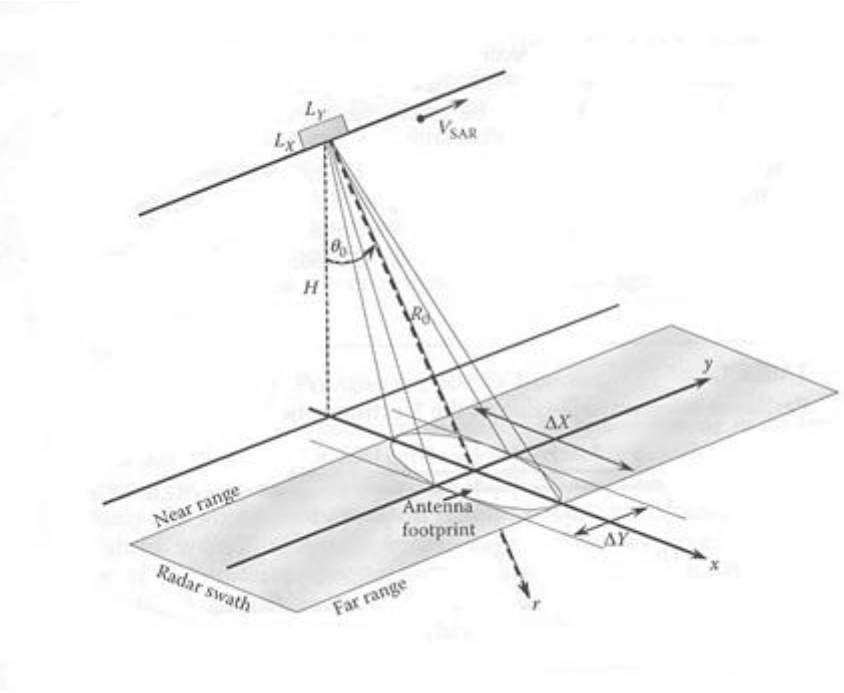


Figure 6. SAR strip-map imaging geometry [Lee & Pottier, 2009].

1.4. SAR Interferometry

SAR interferometry (InSAR) is a remote sensing technique that utilizes the collected phase information of the returned signals over an area, in addition to the amplitude information. The technique utilizes the coherent phase characteristic of a SAR system to derive precise measurements related to the phase delay between the transmitted and received signals. The interferometry process consists of combining the complex images of two different data collections over the same area through a complex cross-correlation. The phase component of this result is called an interferogram and represents the combined phase contributions of the topography, curvature of the earth,

atmosphere, difference in imaging geometry, and deformation. The amplitude component of the cross-correlation result is referred to as a coherence map. The objective of the interferometry is to isolate and systematically account for the contributions of each component to the interferometric phase. This technique is used for generating products such as DEMs as well as monitoring geological processes such as ground deformation.

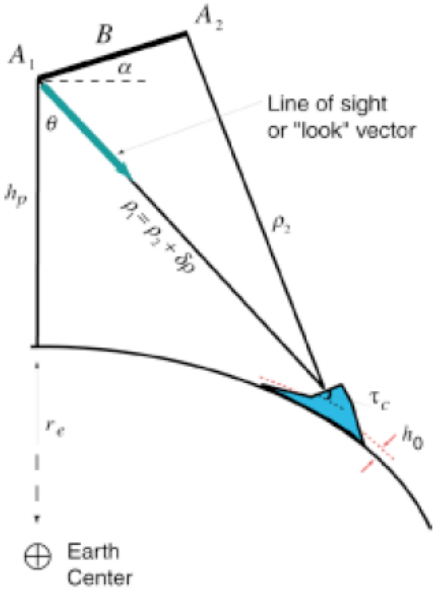


Figure 7. Interferometric SAR imaging geometry [Rosen, 1988].

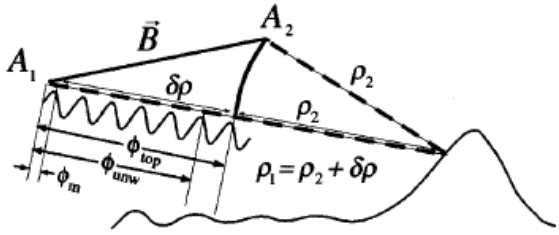


Figure 8. Contributions to differential interferometric phase [Rosen, 1988].

An advanced remote sensing technique known as Polarimetric SAR Interferometry (PolInSAR) combines the techniques of interferometry and polarimetry. By using full polarimetric imagery, the complete scattering behavior of each resolution cell is taken into account in the generation of interferometric products. This allows for the determination of dominate scatterers, accurate estimation of scatterer heights, and the removal of speckle induced errors. Fully polarimetric datasets also allow for advanced techniques such as Polarization Synthesis and Polarimetric Coherence Optimization to be performed prior to interferometric processing.

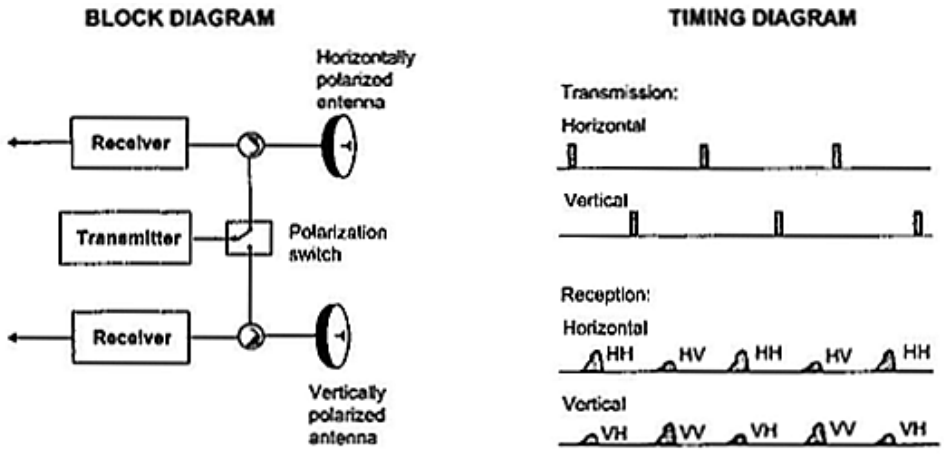


Figure 9. Polarimetric radar operates by using alternating transmit polarizations and simultaneous receive polarizations [Elachi & van Zyl, 2006]

1.5. Organization of Paper

This thesis is organized into a total of five chapters. The current chapter provides an introduction to the concepts and theory as well as background information pertinent to this study. The specifics of the study are detailed in the chapters to follow. Chapter 2 presents the dataset analyzed in this study. It details the area of study, the origin and format of the data, the pre-processing steps applied, and the conventions used throughout the paper. Chapter 3 describes the processing techniques and methodology used for the evaluation of the polarimetric scatterer analysis. The details of each major processing step are described as well as the processing flow. Chapter 4 presents a discussion and summary of the results and findings of the analysis. Chapter 5 summarizes the objectives accomplished and outlines directions of future research to expand upon this work. Additionally, Reference and Appendix sections provide full citations of referenced literature and additional information and data relevant to the processing and analysis of this study.

Chapter 2. Radar Data

2.1. RADARSAT-2

For this study, RADARSAT-2 fully polarimetric datasets were acquired for five dates. RADARSAT-2 is the Canadian Space Agency's (CSA) second-generation commercial SAR satellite, developed in partnership with MacDonald Dettwiler and Associates Ltd. (MDA). It is a multi-mode C-Band imaging sensor designed for earth observation. Operating at a frequency of about 5.405 GHz, RADARSAT-2 orbits the Earth in a sun-synchronous, low Earth orbit at an altitude of 798 km and with an inclination of 98.6 degrees. With its nearly polar inclination, the satellite circles the earth at a rate of 14.3 times a day and has a repeat-pass time over a given area of 24 days. RADARSAT-2 was designed so that it is capable of both right- and left-looking acquisitions as well as a variety of beam modes. The multiple beam modes allow for various combinations of swath widths, resolutions, and polarization options to be achieved.

Table 2. RADARSAT-2 Orbital Parameters [Livingstone, 2005].

Parameter	Value
Orbit type	Sun-synchronous LEO
Altitude	798 km
Inclination	98.6 degrees
Eccentricity	<0.0006
Ascending Node	18:00
Period	100.7 minutes
Orbits per day	14.3
Repeat cycle	24 days

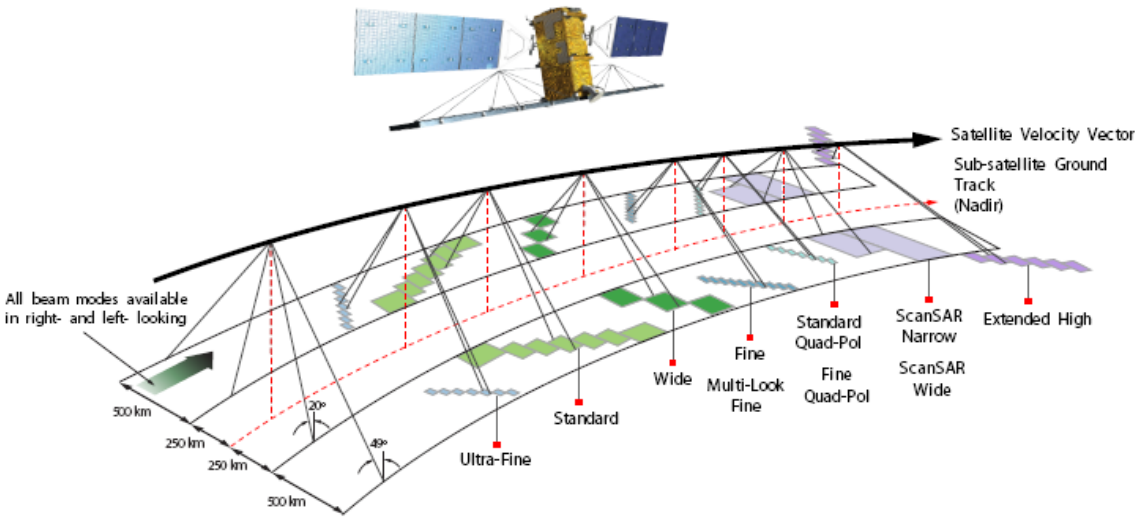


Figure 10. RADARSAT-2 swath coverage for various beam modes [MDA, 2009].

One of the main functional improvements over its predecessor RADARSAT-1 is RADARSAT-2's support of quadrature-polarization (quad-pol) acquisition modes. These quad-pol modes allow for the near-simultaneous collection of imagery using four different transmit-receive polarization configurations, namely HH, HV, VH, and VV. This is achieved by alternately transmitting pulses of H and V polarizations while simultaneously receiving each pulse with both H and V polarizations [Livingstone, 2005]. Because of the increased volume of data in a quad-pol collect, there is a trade off between resolution and swath area. There are two primary quad-pol beam modes, denoted as Standard Quad and Fine Quad.

Table 3. Quad-pol beam mode parameters [MDA, 2009]

Beam Mode	Product	Nominal Pixel Spacing Range x Azimuth [m]	Resolution Range x Azimuth [m]	Nominal Scene Size Range x Azimuth [km]	Range of Angles of Incidence [degrees]	No. of Looks Range x Azimuth
Fine Quad-Pol	SLC	4.7 x 5.1	5.2 x 7.6	25 x 25	20-41	1 x 1
Standard Quad-Pol	SLC	8.0 or 11.8 x 5.1	9.0 or 13.5 x 7.6	25 x 25	20-41	1 x 1

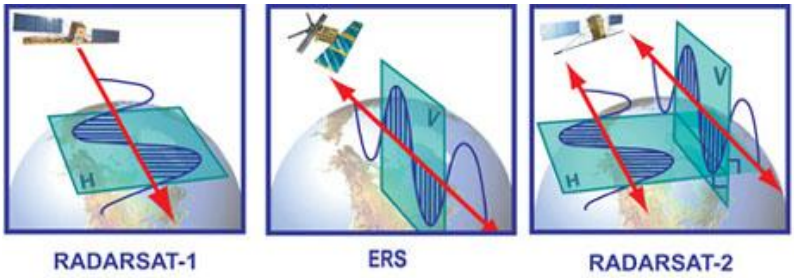


Figure 11. Illustrations of RADARSAT-1, ERS, and RADARSAT-2 polarization configurations [MDA, 2010].

After the RADARSAT-2 satellite acquires data, the data are transmitted a ground station for further processing and archiving. Once the data is back on the ground, the SAR imagery may be processed into various products. For this study, Single Look Complex (SLC) products are utilized. In the SLC format, the imagery preserves both the amplitude and phase data in slant range geometry. Consequently, the SLC product is not geocoded, but instead is georeferenced by its corner coordinates. The quad-pol SLC product consists of the SAR processed images for each supported polarization in separate files [MDA, 2008]. A metadata file containing the acquisition and processing information and calibration look-up tables (LUTs) are also distributed with the imagery.

2.2. RADARSAT-2 SLC Image Stack of Socorro, New Mexico

The image stack consists of five Fine Quad-Pol collections over Socorro, New Mexico. The five collections span a time period of 96 days from June 15, 2008 to September 19, 2008. The Fine Quad-Pol beam mode acquires data over a 25 km by 25 km swath area, with an approximate range resolution of 5.2 meters and an approximate azimuth resolution of 7.6 meters. The specific beam designated for these collections is FQ14, which has an incidence angle of 32.4 degrees.

Table 4. Image parameters of the SLC stack.

Date	Beam Mode	Image ID	Center Coordinates	Start Time
6/15/2008	Fine-Quad Pol, FQ14	PK20649	33.9774966, -106.9650736	1:05:51
7/09/2008	Fine-Quad Pol, FQ14	PK20653	33.9772387, -106.9649361	1:05:52
8/02/2008	Fine-Quad Pol, FQ14	PK20657	33.9772917, -106.9650964	1:05:53
8/26/2008	Fine-Quad Pol, FQ14	PK20661	33.9773164, -106.9650681	1:05:55
9/19/2008	Fine-Quad Pol, FQ14	PK34278	33.9773380, -106.9649043	1:05:58

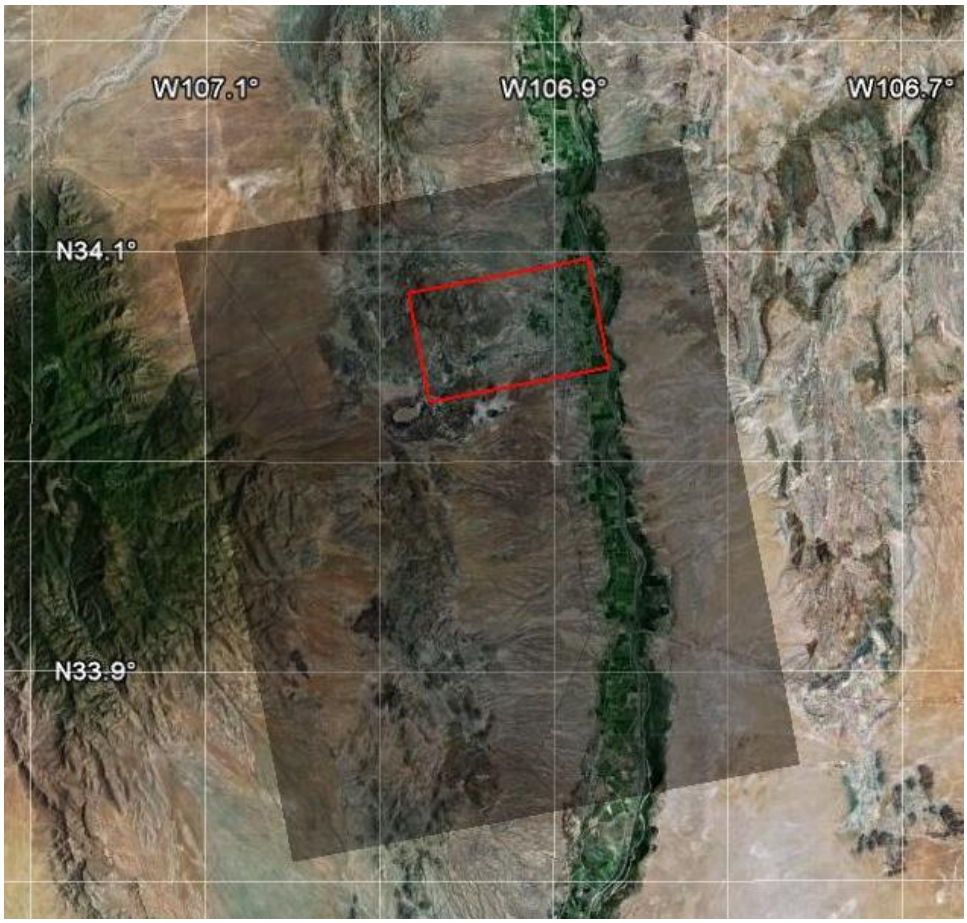


Figure 12. SLC image coverage over Socorro, NM. The shaded region denotes the swath coverage of the image stack while the red outline designates the perimeter of the study area [Google, 2010].

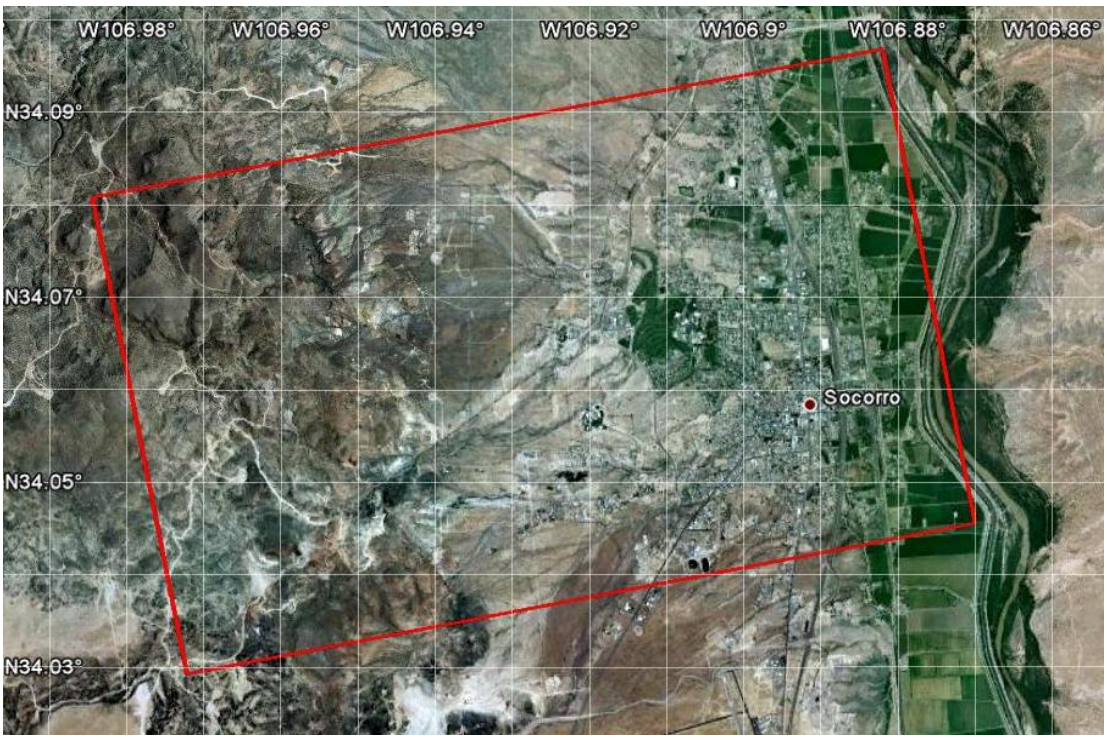
2.3. Data Subset and Conventions

In order to limit volume of data for subsequent processing, the original SLC imagery has been subset to a study area of 1200x1200 pixels as shown in Figure 14. This subset is applied to image stack after coregistration and allows for practical processing constraints and sufficient data for analysis. The study area was selected on the basis of variety of identifiable features and terrain types. It includes a large portion of the city of Socorro as well as the mountains to the West.

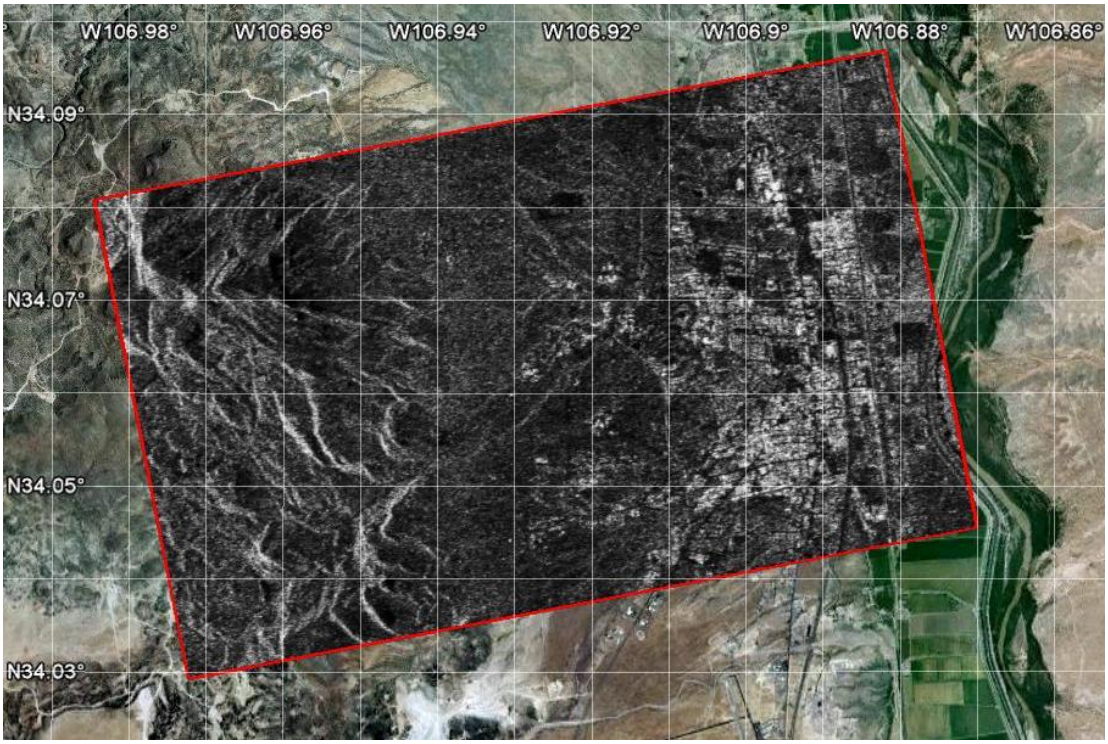
Throughout the remainder of the paper, the image subsets will be specified by acquisition date and polarization. The image acquisition date is presented in YYYYMMDD format. In order for interferometric processing to be performed, all images in the stack are calibrated and co-registered to master slant-range geometry of the 20080802 acquisition. Because of the ascending orbit and look direction of the sensor, the upper-left image coordinate corresponds to the South-West geographic coordinate. Consequently, the images presented in this paper have been flipped vertically to correspond to the conventional cardinal orientation. This is performed as a post-processing step for display and geographic comparison. The relationship between the native image coordinates and the respective geographic coordinates is detailed in Table 4.

Table 5. Image and geographic corner coordinates of the subset area.

Display Location	**Native Image Coordinates** (lines, pixels)	**Geographic Coordinates** (latitude, longitude)
Upper Left	(5901, 1375)	34.08070, -106.98422
Lower Left	(4702, 1375)	34.02947, -106.97216
Upper Right	(5901, 2574)	34.09667, -106.88207
Lower Right	(4702, 2574)	34.04559, -106.87027

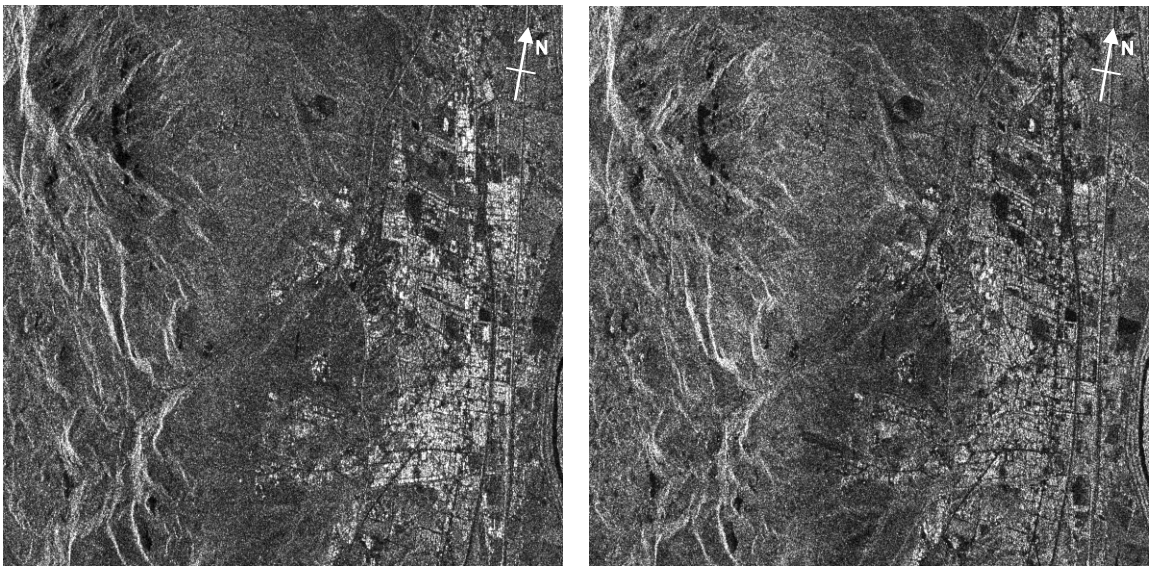

W106.98°
W106.96°
W106.94°
W106.92°
W106.9°
W106.88°
W106.86°
N34.09°
N34.07°
N34.05°
N34.03°
Socorro

(a)

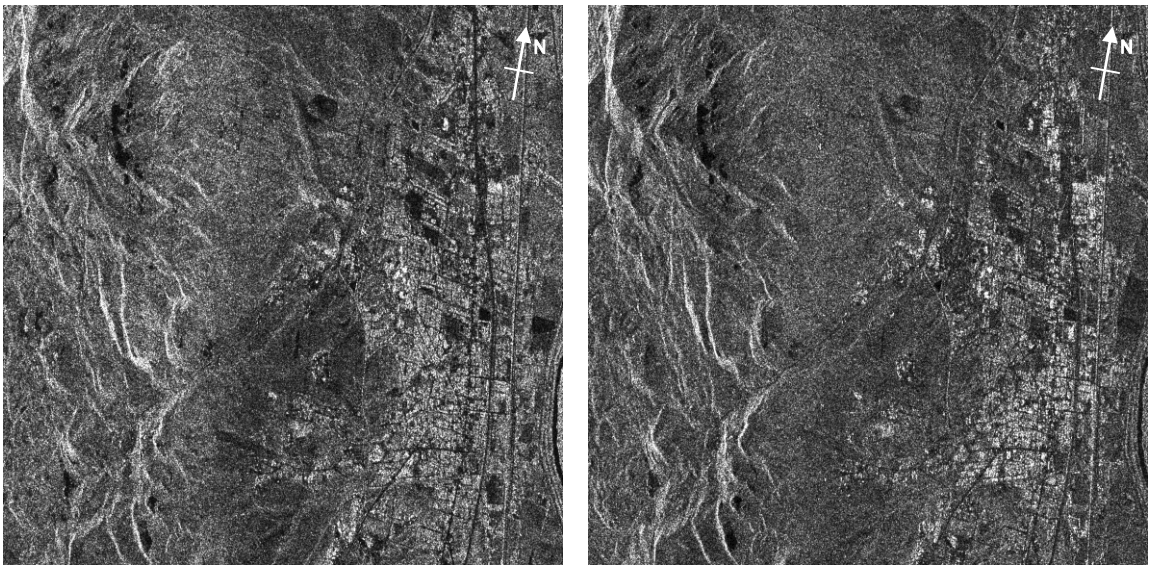


(b)

Figure 13. Ortho-photography of Socorro, NM. The red outline denotes the perimeter of the study area [Google, 2010].



(a) (b)



(c) (d)

Figure 14. SLC intensity of study area from 20080615 quad-pol collect. a) HH. b) HV. c) VH. d) VV.

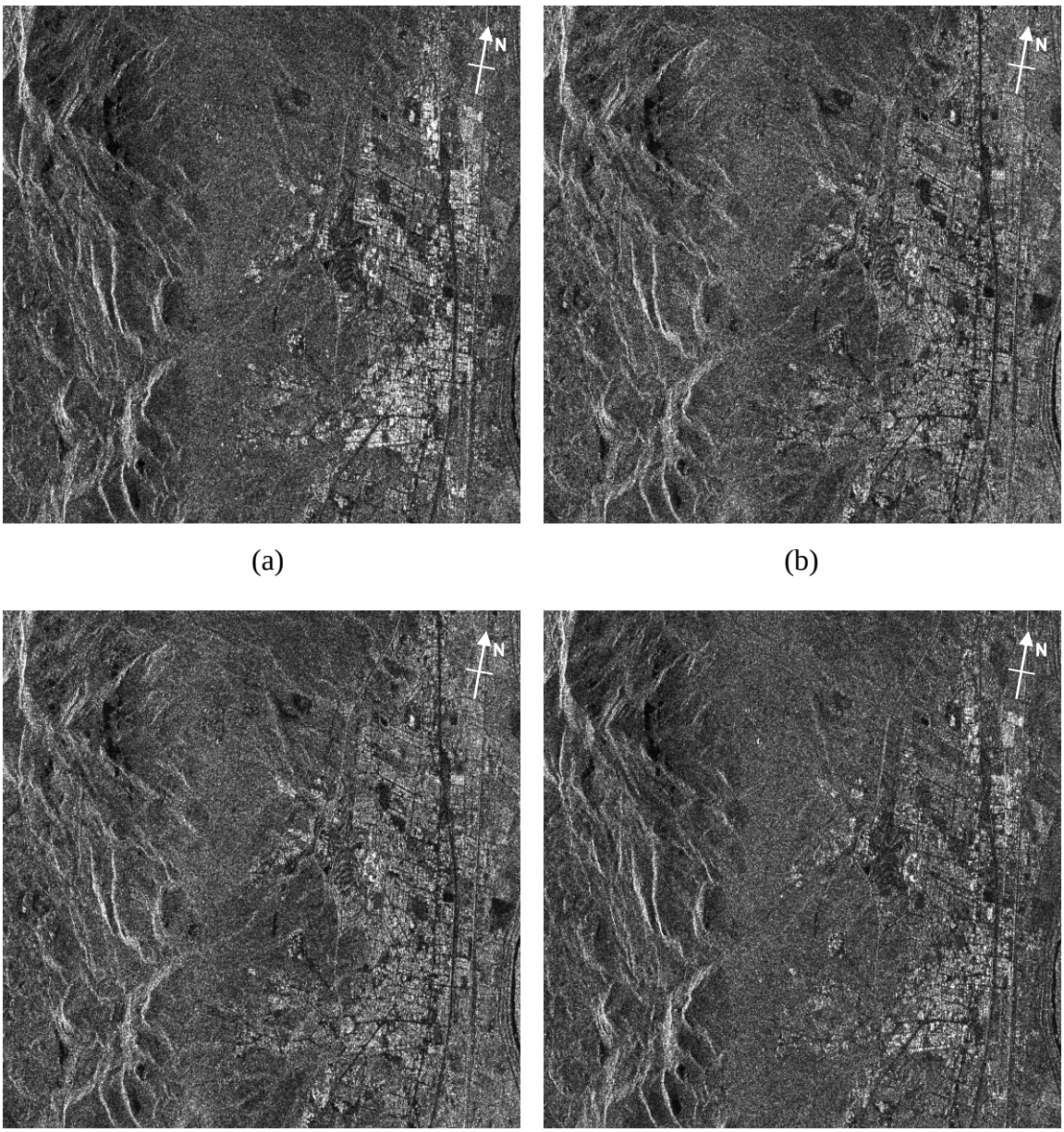


Figure 15. SLC intensity of study area from 20080709 quad-pol collect. a) HH. b) HV. c) VH. d) VV.

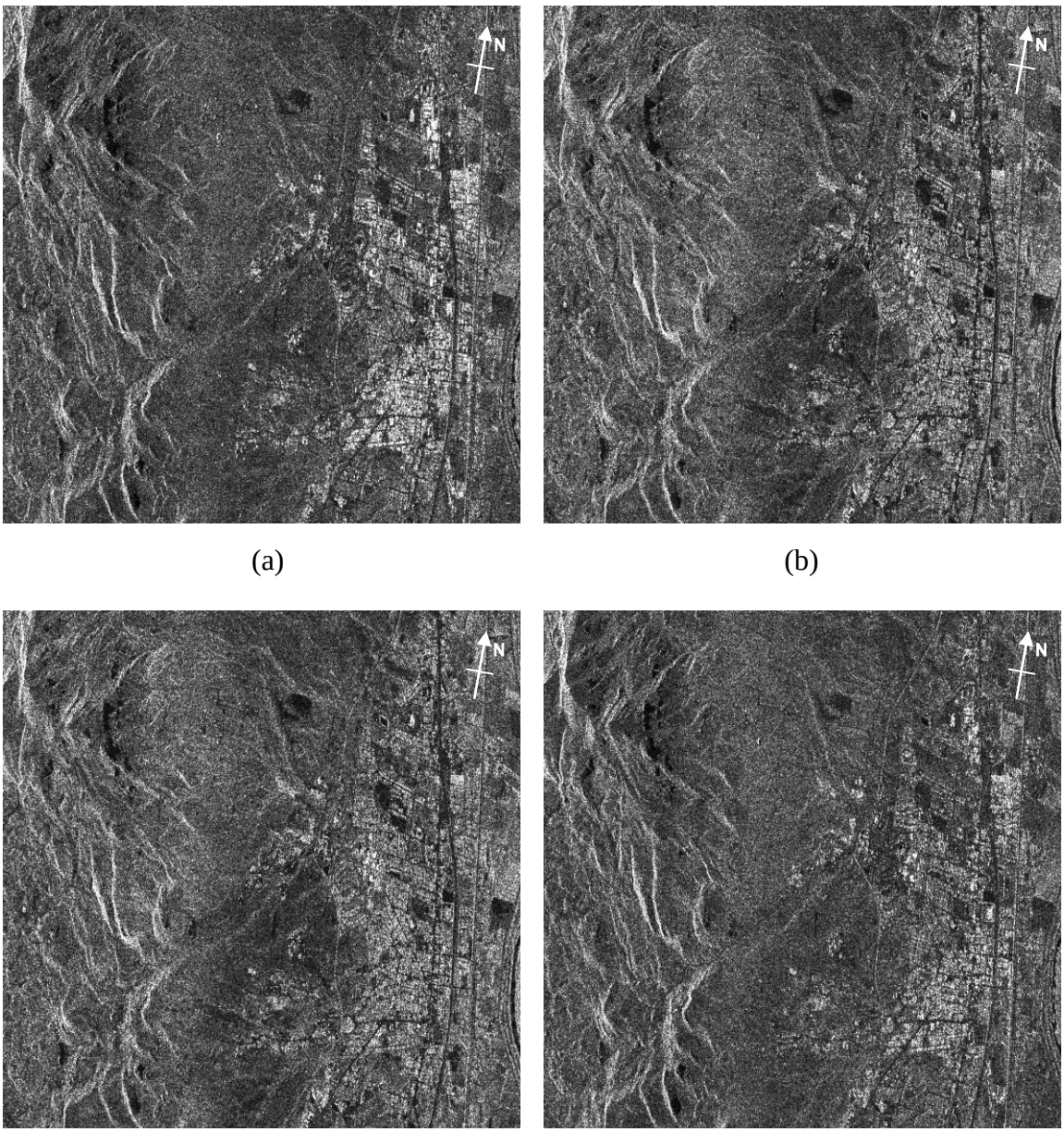


Figure 16. SLC intensity of study area from 20080802 quad-pol collect. a) HH. b) HV. c) VH. d) VV.

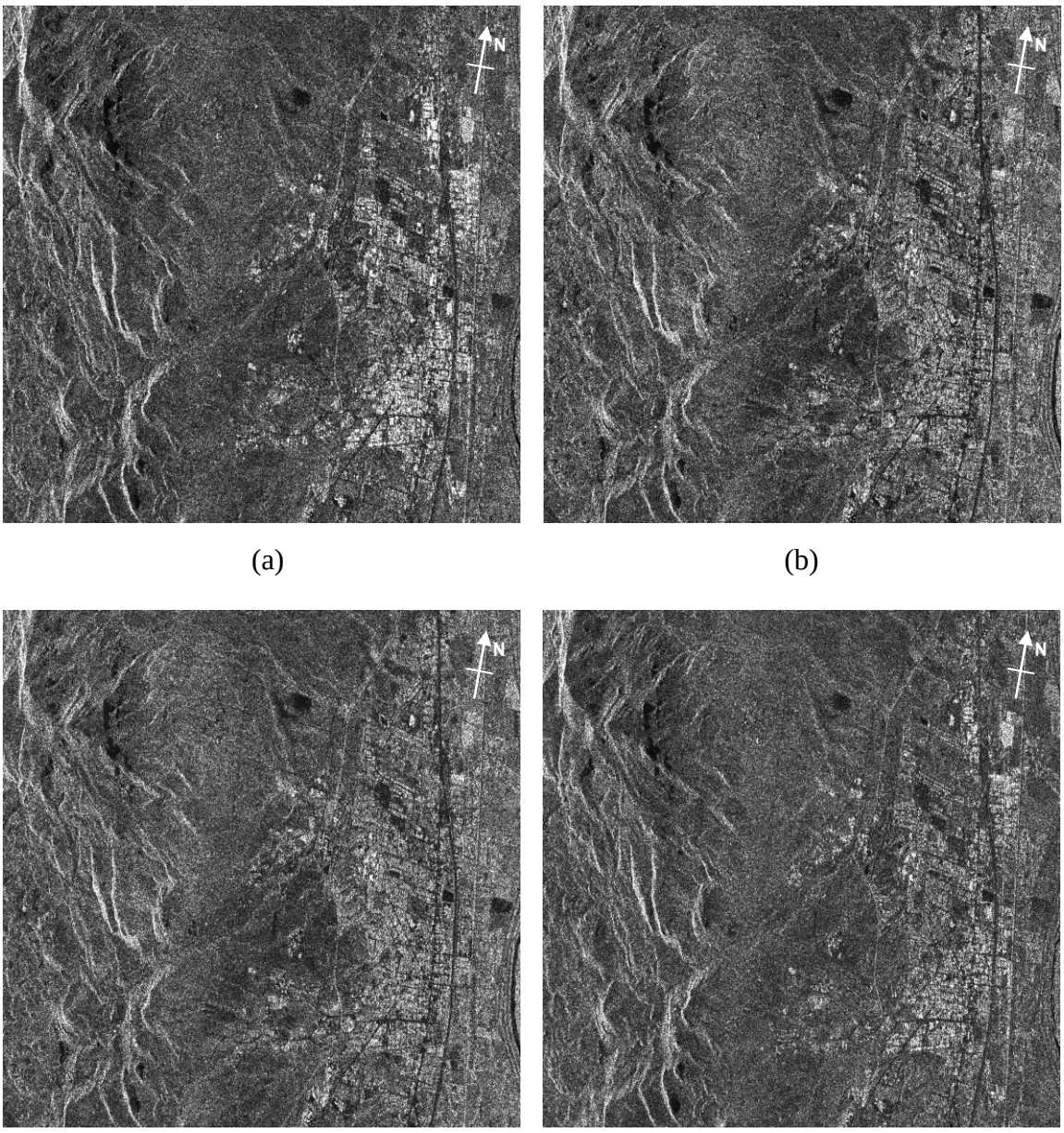


Figure 17. SLC intensity of study area from 20080826 quad-pol collect. a) HH. b) HV. c) VH. d) VV.

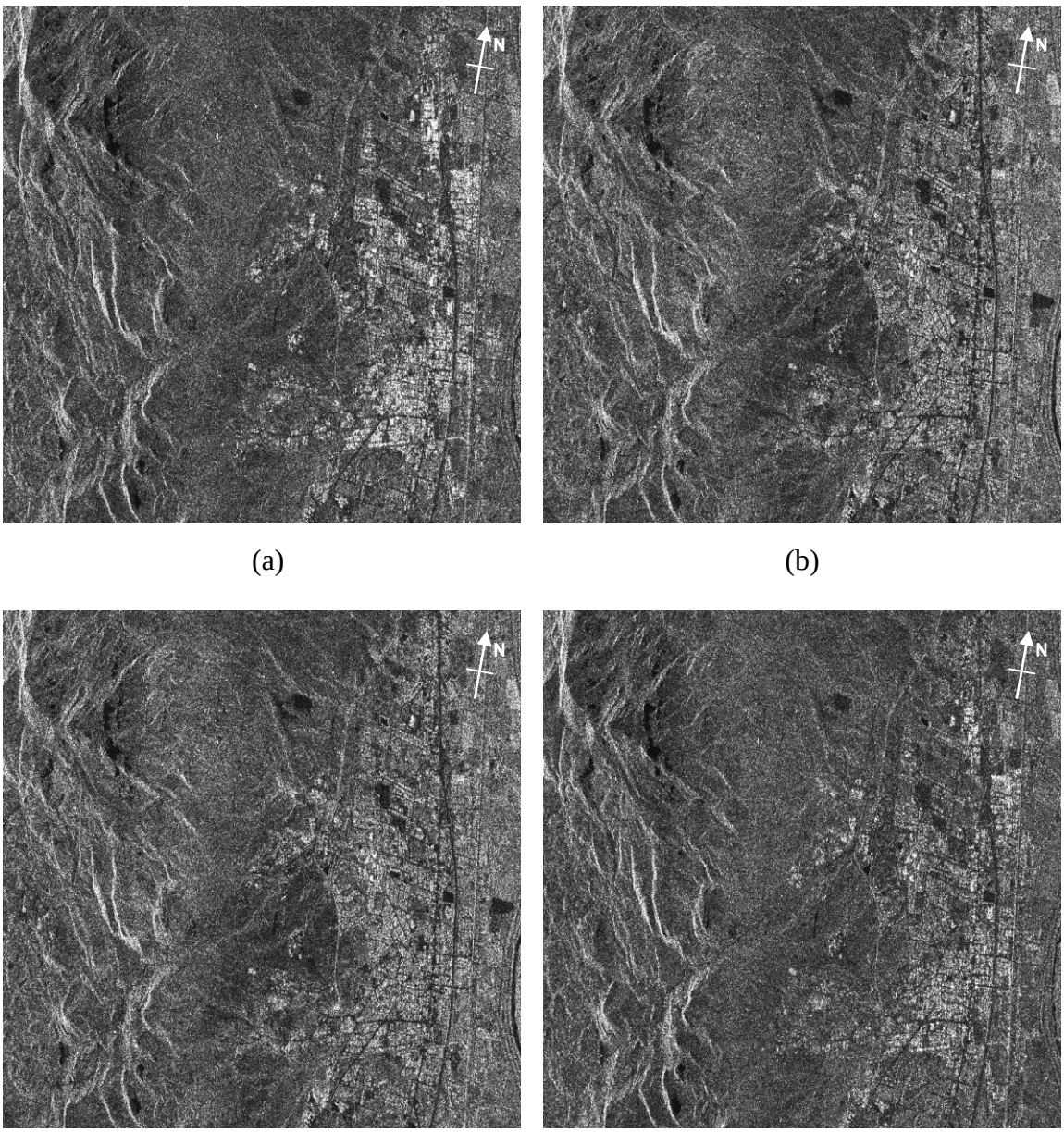


Figure 18. SLC intensity of study area from 20080919 quad-pol collect. a) HH. b) HV. c) VH. d) VV.

2.4. Calibration

As a preprocessing step, each SLC image is corrected for radar brightness. The image calibration performed is a normalization of the radar brightness with respect to the backscatter coefficient, sigma-nought, and is thus referred to as sigma-nought calibration. This calibration takes into account the contribution of the local incidence angle, with respect to the ground plane, to the brightness of a given SAR resolution cell. For RADARSAT-2 data, this correction is calculated at the ground processing station by taking into account system processing parameters and acquisition geometry. A look-up table (LUT) which contains a range dependent gains list is used to scale each resolution cell of the image for the brightness correction [MDA, 2008].

$$calibrated \;\; value = \frac{\left| digital \;\; number \right|^2}{A^2} \tag{2-1}$$

These calibration LUTs are distributed with each SLC imagery product.

2.5. Co-registration

Interferometric SAR processing requires the SLC images to be accurately co-registered to within a fraction of a pixel. The process involves the precise calculation of slant-range offsets in both range and azimuth, and the resampling the images to a master slant-range geometry. By co-registering the image stack to a common geometry, all possible interferometric pairs can be processed and referenced to each other. The sub-pixel accuracy of the image registration also ensures that the interferometric coherence is not degraded due to misalignment of the scatterers. The selection of a master geometry is

made by determining the image acquisition that yields the smallest spatial and temporal baselines between itself and the other images in the stack. The spatial baseline represents the spatial separation between the sensor positions at the times that the images were acquired.

2.6. Digital Elevation Model

A digital elevation model (DEM) is used as an initial model of the study area's topography. The DEM used here is a subset from the National Elevation Dataset (NED) provided by U.S. Geological Survey (USGS) [USGS, 2006]. The NED is primary elevation data product of the USGS. The elevation data are mapped using geographic coordinates in decimal degrees and is referenced to the North American Datum of 1983 (NAD 83). The NED subset that encompasses the study area has a resolution of 1/3 arc-second, which a nominal grid spacing of approximately 10 meters. The elevation values are in units of meters and are referenced to the North American Vertical Datum of 1988 (NAVD 88).

Chapter 3. Radar Processing Methodology

3.1. Overview

The methodology implemented for this study involves the combination of multiple advanced SAR processing techniques. These techniques include polarimetric coherence optimization, interferometric point target analysis, and polarimetric target characterization. In order to perform this processing, several commercial software packages are used. GAMMA software [Gamma, 2008] provides data calibration, coregistration, and point-wise interferometric processing tools as included in the Interferometric Point Target Analysis (IPTA) module. In this study, ENVI's SARscape module by Sarmap [Sarmap, 2009] provides the Polarimetric Coherence Optimization, fully polarimetric interferometry, and polarimetric decompositions. MATLAB software [MathWorks, 2005] provides an environment to develop the additional polarimetric point extraction tools and coherent scatter analysis algorithms. The processing flow and details of each processing step are described in the following sections.

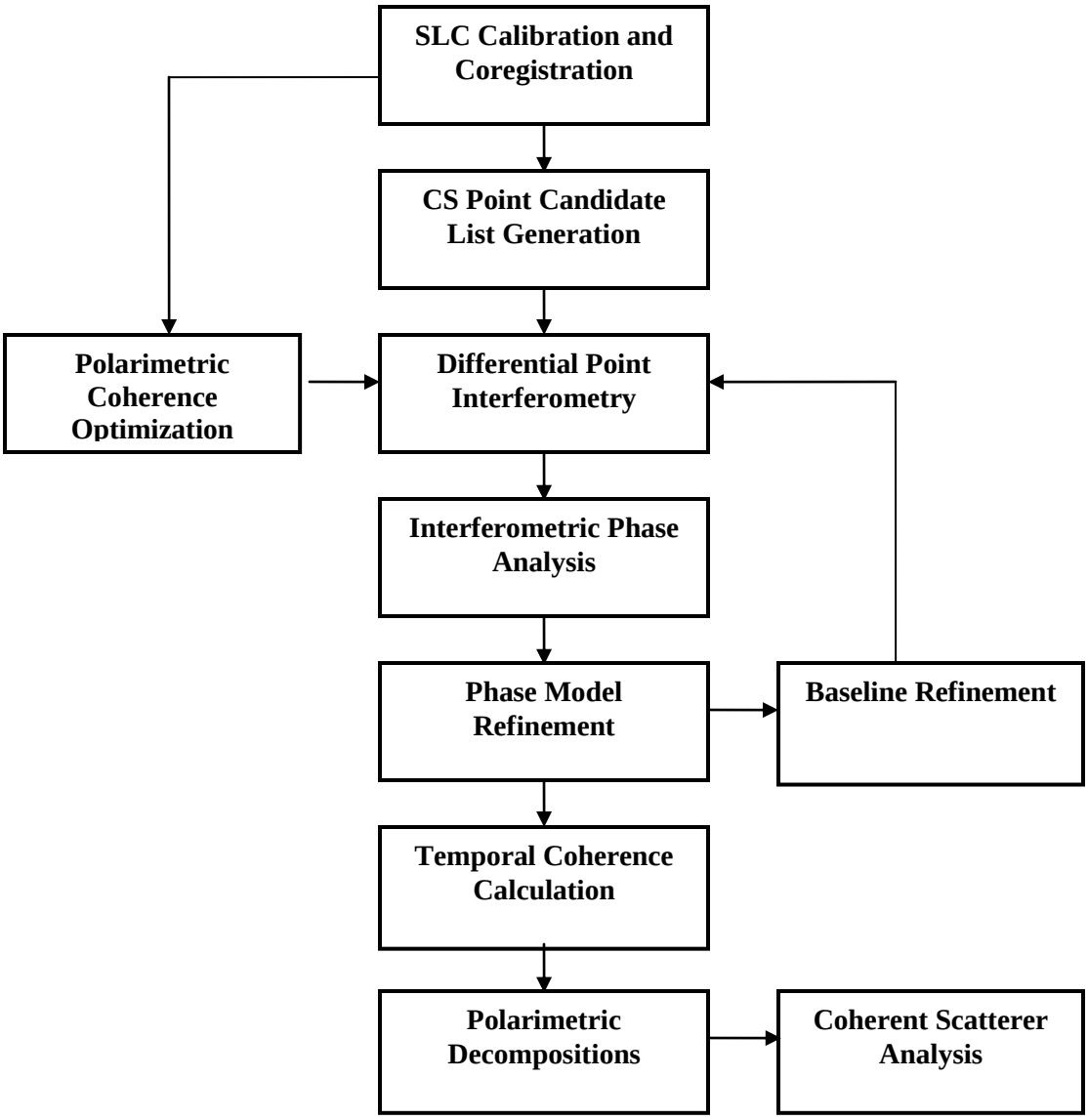


Figure 19. Processing Flowchart.

3.2. Coherent Scatterer Point Candidates

The calculation of the point target candidate list identifies potential point scatterers, or point targets, for analysis and phase model computation. The point list is generated based on measures of low spectral diversity and dominant intensity. Spectral

diversity refers to the characteristic of a point target to dominate the return the relatively larger SAR resolution cell, resulting in a cell that does not exhibit the effects of speckle. The backscattering intensity of resolution cell will therefore remain relatively uniform when compared across several looks. This is evaluated for each individual SLC by generating multiple sub-looks using fractional range and azimuth bandwidths and calculating their spectral correlation. Pixels with a high spectral correlation possess a low measure of spectral diversity. Additionally, the backscattering intensity of a point scatterer is assumed to be dominant when compared to a background level. This characteristic is used to threshold the candidate points to only those that exhibit this dominant intensity [Gamma, 2008]. These measures of spectral diversity and dominant intensity are used in lieu of temporal stability criteria due to the limited number of independent measurements available from the image stack. The point candidate list is further pruned during the model refinement stage.

As described above, the initial point candidates for each individual SLC are determined from spectral characteristics. These point lists derived from the individual SLCs are combined to form the point candidate lists for each of the HH, HV, and VV polarizations. These point lists can then be used to analyze the entire image stack of a given polarization. The point list for a given polarization is determined by retaining only the points that occur in two or more SLC point lists of that polarization. The points that do not meet this constraint are discarded. This assumes that a coherent scatterer at a given location will have enough temporal stability to be present in more than one image in the stack. For the polarimetric coherence optimized case, the point candidate list cannot be formed using the same procedure as the single-pol cases, since there are no

corresponding SLCs available. Instead, the initial point candidate list for the polarimetric coherence optimized case is formed by merging the three single-pol point lists. The initial point lists for each polarization case can be examined in Appendix A.

3.3. Differential Point Interferometry

Since only individual points are being considered for analysis, the differential interferometric calculation is performed on extracted point data rather than on entire images. Once the initial point lists for each polarization case are determined, the SLC data for each point candidate extracted from each SLC of the corresponding polarization. Consequently, this is only performed for the single-pol cases. The extracted SLC values form a point data stack for each polarization. Using these stacks of SLC point values, the complex interferograms are calculated for the HH, HV, and VV polarizations. This is done for all possible interferometric master-slave pairs, yielding a total of ten interferograms for each polarization.

Table 6. Interferometric pairs.

Stack Layer	Master Image	Slave Image	Perpendicular Baseline [m]
1	20080802	20080615	134.8703
2	20080802	20080709	113.1424
3	20080802	20080826	214.3413
4	20080802	20080919	346.5377
5	20080615	20080709	21.7279
6	20080615	20080826	349.2116
7	20080615	20080919	481.4080
8	20080709	20080826	327.4837
9	20080919	20080709	459.6801
10	20080919	20080826	132.1964

As done for the SLCs, height data for each point list is extracted from the slant-range DEM. From this, the simulated unwrapped interferometric phase is calculated for each case. The simulated, or synthetic, unwrapped phase is derived from the phase contributions of the acquisition geometry of each interferometric image pair as well as the topography. This simulated unwrapped phase is subtracted from the complex interferograms to generate the differential interferogram point data stacks. These differential interferograms are subsequently analyzed using phase regression analysis and are included in Appendix C for reference.

3.4. Polarimetric Coherence Optimization

With this study's fully polarimetric RADARSAT-2 data, polarimetric coherence optimization is utilized in the interferometric processing. The interferometric coherence may vary with the polarization. Thus, it is of interest to maximize the coherence using the polarimetric data. By calculating the deterministic scattering mechanism that most closely represents the contribution of a point-scatterer for a given resolution cell, the optimum coherence map can be generated. For this study, the Cloude and Papathanassiou polarimetric coherence optimization algorithm [Cloude, 1998] is implemented by SARscape's Polarimetric Coherence Optimization routine, which generates the optimized interferograms and coherence maps for the quad-pol datasets. These polarimetric coherence optimized interferograms are ingested into the target analysis and compared against that from the single-pol case.

In order to utilize the information provided by a fully polarimetric dataset, the conventional interferometric formulas must be generalized to allow for vector

interferometric calculations. This is performed by first transforming the scattering matrix in a vector form. The resulting vector is referred to as the scattering vector. Assuming reciprocity and applying the Pauli decomposition [Lee and Pottier, 2009], the scattering matrix

$$[S]=\begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \tag{3-1}$$

becomes the scattering vector

$$\vec{k}=\frac{1}{\sqrt{2}}\begin{bmatrix} S_{HH}+S_{VV} & S_{HH}+S_{VV} & 2S_{HV} \end{bmatrix}^T \tag{3-2}$$

Then using vector interferometry, the generalized coherence formula becomes

$$\gamma=\frac{\left|\left\langle \vec{\omega}_1^{*T}[\Omega_{12}]\vec{\omega}_2 \right\rangle\right|}{\sqrt{\left\langle \vec{\omega}_1^{*T}[T_{11}]\vec{\omega}_1 \right\rangle\left\langle \vec{\omega}_2^{*T}[T_{22}]\vec{\omega}_2 \right\rangle}} \tag{3-3}$$

where

$$[T_{11}]=\left\langle k_1 k_1^{*T} \right\rangle \tag{3-4}$$

$$[T_{22}]=\left\langle k_2 k_2^{*T} \right\rangle \tag{3-5}$$

$$[\Omega_{12}]=\left\langle k_1 k_2^{*T} \right\rangle \tag{3-6}$$

and $\vec{\omega}_1$ and $\vec{\omega}_2$ are normalized complex vectors that represent the selected scattering mechanisms [Sarmap, 2009]. This generalized form of the coherence represents the contributions of both interferometric and polarimetric coherence.

In order to evaluate the coherence due to different scattering mechanisms, it is necessary to perform polarization basis transformations. Polarization basis transformation is the process of converting the scattering vector from one orthogonal basis to another.

This is possible if a scattering vector in any one orthogonal basis is known [Lee and Pottier, 2009]. In this case, since the scattering vector is known for a given resolution cell in the (H,V)-basis, it can be calculated in any orthogonal (A,B)-basis. Effectively, this allows for interferogram and coherence calculation with respect to different selected scattering mechanisms as

$$\vec{k}_{AB} = [U_3]\vec{k}_{HV} \tag{3-7}$$

$$[U_3] = \frac{1}{2(1+\rho\rho^*)}\begin{bmatrix} 2+\rho^2+\rho^{*2} & \rho^{*2}-\rho^2 & 2(\rho-\rho^*) \\ \rho^2-\rho^{*2} & 2-(\rho^2+\rho^{*2}) & 2(\rho+\rho^*) \\ 2(\rho-\rho^*) & -2(\rho+\rho^*) & 2(1-\rho\rho^*) \end{bmatrix} \tag{3-8}$$

where the polarization ratio, ρ, is defined as

$$\rho = \tan(\alpha)e^{j\theta} \tag{3-9}$$

With the ability to evaluate different scattering mechanisms by performing polarimetric basis transformations, the coherence of each polarization state may be found for each resolution cell.

This optimization problem is formulated by maximizing the complex Lagrangian L formed from the general vectorized coherence formula as

$$L = \vec{\omega}_1^{*T}[\Omega_{12}]\vec{\omega}_2 + \lambda_1(\vec{\omega}_1^{*T}[T_{11}]\vec{\omega}_1 - C_1) + \lambda_2(\vec{\omega}_2^{*T}[T_{22}]\vec{\omega}_2 - C_2) \tag{3-10}$$

This becomes an eigenvalue problem, such that maximum coherence value is the square root of the maximum eigenvalue. From this maximum coherence value, the two optimum scattering mechanisms are determined. The result of the coherence optimization process is three sets of flattened interferograms and coherence maps. Each interferogram and coherence map pair corresponds to one of the three singular values yielded by the solution to the optimization problem [Cloude, 1998].

In this study, of the three flattened interferograms produced for each master-slave pair, only the interferogram corresponding to the optimal singular value is used for subsequent processing. In order to integrate the polarimetric coherence optimized (PCO) interferograms into the point target analysis process, the corresponding point candidate list is used to extract the point data from the optimized differential interferograms. Because the SARscape and GAMMA software packages calculate the simulated interferometric phase differently, the simulated phase that was used for coherence optimization process is added back to the resulting differential interferogram. This yields PCO complex interferograms and allows the GAMMA simulated phase to be used for flattening. This is necessary for the PCO data and the single-pol data to be handled consistently during the interferometric point analysis and model refinement steps.

3.5. Interferometric Point Analysis and Model Refinement

With differential interferogram point data stacks for the four polarization cases, a model of the interferometric phase is developed to account for variations across the layers of the stack. However, before this is performed, a point-wise minimum cost flow unwrapping process is applied to the differential interferograms. This a priori spatial unwrapping avoids the potential for unwrapping errors during the phase regression analysis due to the relatively short image stack. The phase regression analysis assumes that the difference in phase between two points in the interferogram is linearly related to the perpendicular baseline component of the topography. Deviation of the phase values from this linear regression fit is attributed noise, atmospheric, and baseline error

contributions [Gamma, 2008]. Due to the temporal coverage of the image stack it can be assumed that there is no significant deformation present and therefore linear deformation rate is not considered in the phase model calculation for this study. The result of the phase regression analysis is a set of height corrections, residual phases, point quality measures, and resulting unwrapped interferometric phase for each layer of the point data stack.

After regression analysis, the height corrections are used to refine the height model, which was initially derived from the DEM. Additionally, the point quality measures are used prune the point lists, retaining only points above a certain threshold for further inclusion in the analysis. Following the initial phase regression analysis and model refinement, the differential interferograms are recalculated using the updated height maps and point lists. This sequence is an iterative process and can be carried out multiple times to increase the quality of the phase model. For this study the sequence is performed twice. After the first iteration, the updated height maps and unwrapped interferometric phase are used for baseline refinement. Baseline refinement is a least squares estimation technique that produces higher quality baselines than those initially derived from orbital parameters [Gamma, 2008]. Using the improved baselines, the differential interferogram point data is recalculated and a second iteration of the regression analysis and model refinement is performed. The resulting differential interferograms are then used for further analysis. These interferograms are included in Appendix D.

3.6. Temporal Coherence Estimation

Once the interferometric point analysis is completed for each polarization case, the temporal phase coherence of the points in each of the four resulting point data stacks is estimated. This is done using the Gamma software package. The temporal coherence is the correlation of the interferometric point data for each pixel, between layers of the data stack. It is calculated for each point as the magnitude of the sum of the interferometric values from each layer, normalized by the number of layers in the stack.

$$\gamma_T = \frac{1}{N}\left|\sum_{n=1}^{N} A_n e^{j(\theta_n - \phi_n)}\right| \tag{3-11}$$

In order to calculate this value, it is necessary to determine the local average phase value (ϕ_n) for each layer, for a given point, and subtract it from the point's phase value in the corresponding layer. This is accomplished by using a weighted average of the point values within a fixed spatial distance from the point being considered. If there is not a sufficient number of neighboring points for this weighted average, local average phase for the particular point is not calculated. The result is that the temporal coherence is not necessarily calculated for all points in a data stack [Gamma, 2008]. For this study, a spatial search distance of 8 pixels and a minimum number of 5 neighbors are used for the estimation. The temporal coherence maps provide a measure of the stability of the interferometric phase values for each point and are further used for polarimetric scatterer analysis.

3.7. Polarimetric Decompositions - Pauli Method

To characterize the polarimetric behavior of the point scatterers in the final point lists, two polarimetric decomposition methods are employed. The first is the Pauli decomposition. The Pauli decomposition is a coherent polarimetric decomposition method. It expresses the scattering matrix in terms of simpler canonical scattering models. These canonical models represent independent scattering mechanisms. The scattering matrix can be written as the weighted sum of the responses from the canonical models.

$$[S]=\begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} = \sum_{i=1}^{k} c_i [S]_i = \alpha [S]_a + \beta [S]_b + \gamma [S]_c$$

(3-12)

For the reciprocal case of a monostatic radar system, the Pauli decomposition is comprised of three scattering models,

$$[S]_a = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3-13}$$

$$[S]_b = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{3-14}$$

$$[S]_c = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{3-15}$$

The scattering matrix then becomes

$$[S]= \alpha [S]_a + \beta [S]_b + \gamma [S]_c \tag{3-16}$$

such that

$$\alpha = \frac{S_{hh} + S_{vv}}{\sqrt{2}} \tag{3-17}$$

$$\beta = \frac{S_{hh} - S_{vv}}{\sqrt{2}} \tag{3-18}$$

$$\gamma = \sqrt{2} S_{hv} \tag{3-19}$$

The canonical scattering matrices $[S]_a$, $[S]_b$, and $[S]_c$ physically correspond to the scattering responses from odd-bounce, even-bounce, and a dihedral rotated to a 45 degree orientation. The Pauli decomposition is performed using the SARscape Polarimetry module. In this study, classification based on the Pauli decomposition is performed by determining the scattering component with the largest amplitude. Due to the short temporal span of the data in this study, the polarimetric decompositions were performed on the geometric master only, the 20080802 collect.

3.8. Polarimetric Decompositions - Alpha/Entropy Method

In addition to the Pauli decomposition, the polarimetric scattering characteristics of the final point lists are also examined in terms of the Alpha/Entropy decomposition [Cloude and Pottier, 1997]. For the Alpha/Entropy method, a dominant or average scattering mechanism is assumed for each location, which may be described by two parameters, scattering entropy and alpha angle. Scattering entropy (H) is a measure of the roughness and polarization dependence of the scatterer. The alpha angle (α) correlates with a pixel's average scattering mechanism [Cloude and Pottier, 1997]. From the 2 x 2 scattering matrix **S**, the coherency matrix **T** is calculated as

$$\langle [T] \rangle = \frac{1}{2} \langle \begin{bmatrix} (S_{HH}+S_{VV})(S_{HH}+S_{VV})^* & (S_{HH}+S_{VV})(S_{HH}-S_{VV})^* & 2(S_{HH}+S_{VV})S_{HV}^* \\ (S_{HH}-S_{VV})(S_{HH}+S_{VV})^* & (S_{HH}-S_{VV})(S_{HH}-S_{VV})^* & 2(S_{HH}-S_{VV})S_{HV}^* \\ 2S_{HV}(S_{HH}+S_{VV})^* & 2S_{HV}(S_{HH}-S_{VV})^* & 4S_{HV}S_{HV}^* \end{bmatrix} \rangle$$

(3-11)

Next, Eigen decomposition [Wesstein, 2002] of **T** yields

$$\langle [T] \rangle = [U_3] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} [U_3]^{*T} \text{ where } \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0 \quad (3\text{-}12)$$

as in Sublook Entropy, and $[U_3]$ contains eigenvectors $[e_i]$ as its columns. Each column eigenvector has the form

$$[e_i] = \begin{bmatrix} \cos \alpha_i \\ \sin \alpha_i \cos \beta_i \exp(j\delta_i) \\ \sin \alpha_i \sin \beta_i \exp(j\gamma_i) \end{bmatrix} \quad (3\text{-}13)$$

so that α_i may be recovered as $\arccos\left(\sqrt{[e_i]_1 [e_i]_1^*}\right)$ and

$$p_i = \frac{\lambda_i}{\sum_{j=1}^{N} \lambda_j}. \quad (3\text{-}14)$$

Finally, entropy and alpha are formed

$$\bar{\alpha} = \sum_{i=1}^{3} p_i \alpha_i \quad (3\text{-}15)$$

$$\bar{H} = \sum_{i=1}^{3} p_i \log_3(p_i) \quad (3\text{-}16)$$

The α-H plane is illustrated in Figure 20.

Z1: High Entropy Multiple Scattering
Z2: High Entropy Vegetation Scattering
Z3: High Entropy Surface Scatter
Z4: Medium Entropy Multiple Scattering
Z5: Medium Entropy Vegetation Scatter
Z6: Medium Entropy Surface Scatter
Z7: Low Entropy Multiple Scattering
Z8: Low Entropy Dipole Scattering
Z9: Low Entropy Surface Scatter

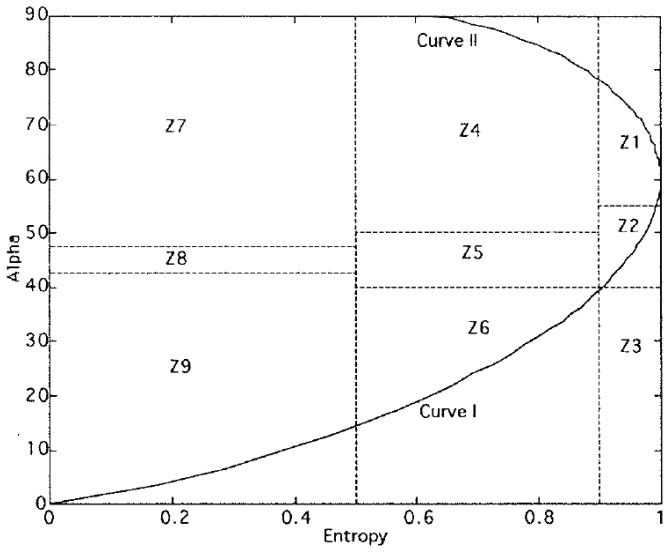


Figure 20. The α-H plane [Cloude and Pottier, 1997].

3.9. Coherent Scatterer Point List Analysis Methodology

To examine the polarimetric characteristics and interferometric phase quality of the point targets, the information from the temporal coherence maps and polarimetric decompositions are analyzed. Since this involves

The quality of the point targets used for particular polarization can be compared to that of another be evaluating the statistical distribution of the temporal coherence for the point data stacks of each polarization. A histogram of the coherence values as well as first order statistics are calculated for each of the four polarization cases. In order to investigate the relation between interferometric phase quality and scatterer classification, the temporal coherence distribution computed for each scatterer type of both the Pauli and α-H methods.

Chapter 4. Processing Results and Discussion

4.1. Introduction

Interferometric Synthetic Aperture Radar (InSAR) provides a means of imaging small deformations of the Earth's surface (millimeter scale). Interferometric point target analysis (IPTA) and time-series analyses can be enhanced using polarimetric and interferometric synthetic aperture radar (PolInSAR) processing techniques to improve detection of point targets or coherent scatterers (CS) that are characterized by a point-like scattering behavior.

For enhancing InSAR techniques, this study explores procedures and parameters for detecting polarimetric coherence optimized CS candidates using InSAR IPTA. In this chapter, we assess the enhancement in CS point identification using full polarization for the Socorro, New Mexico site. Analysis of the CSs is performed on the results for examining the potential benefits of using the full polarization compared to without full polarization.

4.2. Coherent Scatterer Point Locations

The CS results from the study site data are examined. Following the chapter 3 methodology, CS lists were outputted from IPTA processing of the CS candidates for HH, HV, VV, and PCO. A summary of the CS results are shown in Table 7. The IPTA output differential interferograms are shown in the Appendices. For the HH, HV, VV,

and full polarization, an example differential interferogram from the study site is shown in Figure 21. Clearly, the full polarization shows denser CS population.

Comparison of the full polarization PCO result to HH, HV, and VV polarizations was conducted. Overlap amongst the single polarization CS lists was small (<8%). In contrast, the vast majority (>97%) of the HH, HV, and VV polarization CSs overlap with full polarization PCO CS list. On a single polarization basis, the non-overlap CSs (<3%) are likely due to false alarm CS detections. Thus, the full polarization CS list contains more than 2.5 times the largest single polarization CSs from the HH single polarization.

For full polarization, the increase (>250%) in CSs has a number of potential benefits. This significant increase in full polarization CSs may increase the overall InSAR processing quality. For sites that challenge (e.g. vegetated areas and other low coherence environments) the single polarization InSAR processing, the full polarization increase in CSs may be essential for processing. A larger number of images and quantity of geographic sites then were available for this study is needed to further quantify.

Table 7. InSAR IPTA CS output lists are summarized. Full polarization PCO CS list compared with HH, HV, and VV polarization lists. Image contains 1,440,000 pixels (1200 rows by 1200 columns).

Polarization	CS	CS Overlap with PCO
HH	23,951	23,429
HV	19,273	18,910
VV	24,124	23,421
PCO	61,884	61,884

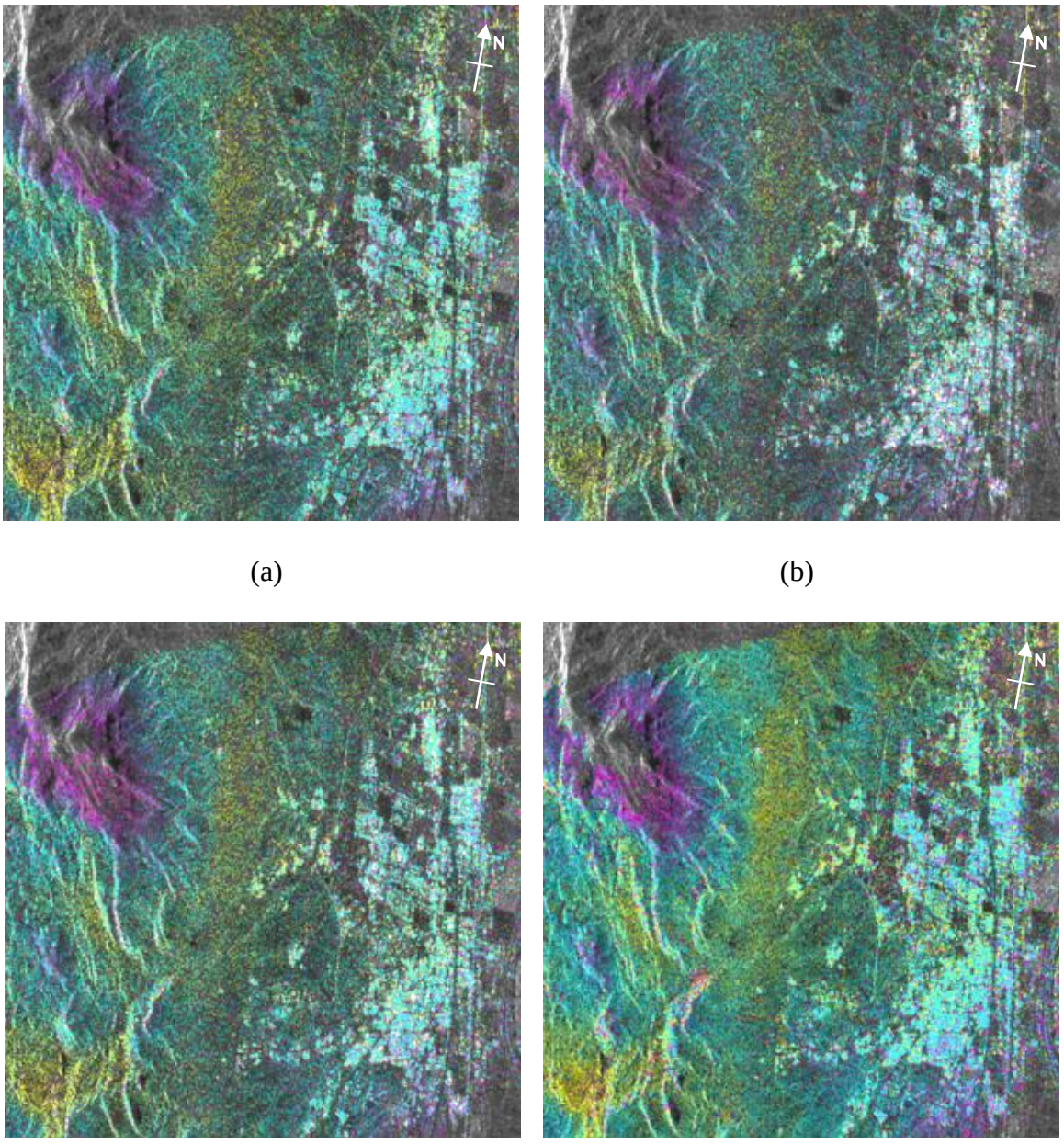


Figure 21. Differential interferograms 20080709_20080826 are shown for each polarization: a) HH, b) HV, c) VV, and d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

4.3. Polarimetric Scattering Behavior

The polarimetric scattering behaviors of the CS lists were investigated using the Alpha-Entropy and Pauli decompositions, which were described in Chapter 3. Using the scattering decompositions, classification maps were generated as shown in Figure 22. Int the classification image, colors approximately correspond as red to multiple scattering, green to volume scattering, and blue to surface scattering. To explore the scattering behavior of the HH, HV, VV, and PCO CS lists, the decomposition classification of each pixel in the CS lists were tabulated as shown in Table 7.

For the study site, the CSs were dominated in each decomposition case. For Alpha-Entropy, zones 6 and 9 correspond to surface scattering mechanisms. HH, HV, VV, and PCO CS lists contain about 71.6%, 59.0%, 73.2%, and 68.4% CS points classified within the Alpha-Entropy zones 6 and 9. The CS lists each corresponded to about 39% of Pauli odd bounce parameter K1, which approximately corresponds to a physical interpretation as a combination of surface, sphere, corner reflector scattering. For the CSs, the statistics (i.e. mean, variance, and standard deviation) for each class type were not significantly different. Thus, the Pauli scattering statistics for the CSs were consistent with no significant variation amongst each CS list. The study site is not geographically dominated with man-made structures. Thus, the decomposition results seem to align qualitatively well with the local geography in scattering mechanism.

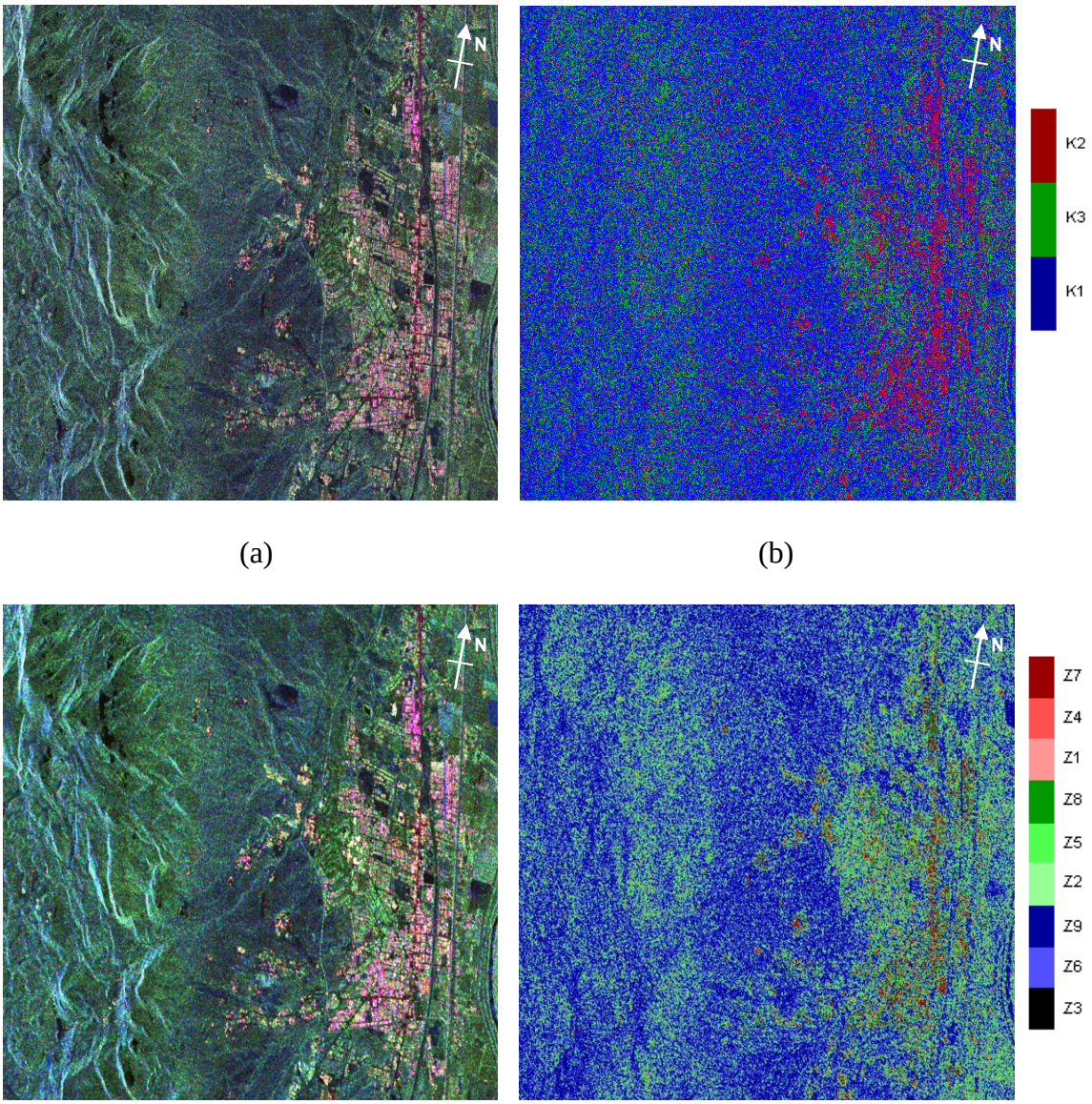


Figure 22. A decomposition summary of the study site is shown. a) Pauli decomposition. b) Pauli classification map. c) α-H decomposition. d) α-H classification map. Master image is 20080802.

Table 8. Pauli decomposition and classification statistics by point list and scatter type.

CS List Polarization	Pauli Component	Number of Points	Component Statistics		
			μ	σ^2	σ
HH	**K1**	**9,497**	0.0345	0.0007	0.0256
	K2	**7,498**	0.0228	0.0008	0.0281
	K3	**7,149**	0.0143	0.0001	0.0103
HV	**K1**	**7,569**	0.0345	0.0007	0.0262
	K2	**5,916**	0.0213	0.0006	0.0253
	K3	**5,662**	0.0145	0.0001	0.0103
VV	**K1**	**9,483**	0.0344	0.0007	0.0257
	K2	**7,488**	0.0219	0.0008	0.0281
	K3	**7,041**	0.0145	0.0001	0.0106
PCO	**K1**	**24,387**	0.0339	0.0006	0.0251
	K2	**19,152**	0.0212	0.0006	0.0255
	K3	**18,345**	0.0143	0.0001	0.0101

Table 9. α-H decomposition and classification statistics by polarization and scatterer type.

CS List Polarization	α-H Zone	Number of Points	α Component Statistics			H Component Statistics		
			μ	σ^2	σ	μ	σ^2	σ
HH	**Z1**	**2**	60.8815	25.5306	5.0228	0.9177	0.0001	0.0108
	Z2	**24**	50.9354	4.8325	2.1983	0.9198	0.0002	0.0139
	Z3	**0**	---	---	---	---	---	---
	Z4	**640**	60.1955	18.5344	4.3052	0.6486	0.0099	0.0997
	Z5	**3,872**	45.8248	15.9750	3.9969	0.6728	0.0081	0.0899
	Z6	**6,292**	32.7620	20.5849	4.5371	0.5918	0.0043	0.0654
	Z7	**1,111**	65.9777	56.0901	7.4893	0.2961	0.0146	0.1209
	Z8	**1,200**	46.7515	18.3684	4.2858	0.3620	0.0125	0.1119
	Z9	**11,003**	23.9150	47.8181	6.9151	0.3582	0.0089	0.0946
HV	**Z1**	**13**	58.5530	9.9097	3.1480	0.9219	0.0002	0.0133
	Z2	**47**	51.0515	5.8392	2.4165	0.9215	0.0004	0.0196
	Z3	**0**	---	---	---	---	---	---
	Z4	**945**	60.3345	19.4809	4.4137	0.6810	0.0103	0.1013
	Z5	**5,269**	46.2065	16.3452	4.0429	0.6892	0.0082	0.0907
	Z6	**5,481**	33.6899	18.6435	4.3178	0.6010	0.0046	0.0682
	Z7	**697**	65.4897	50.5179	7.1076	0.3280	0.0153	0.1235
	Z8	**878**	46.1894	17.9237	4.2336	0.3932	0.0089	0.0942
	Z9	**5,817**	26.0848	44.1797	6.6468	0.3822	0.0074	0.0862
VV	**Z1**	**3**	55.8568	0.8719	0.9338	0.9258	0.0007	0.0269
	Z2	**14**	51.7608	4.1357	2.0336	0.9189	0.0002	0.0147
	Z3	**0**	---	---	---	---	---	---
	Z4	**644**	60.6188	21.8786	4.6775	0.6478	0.0094	0.0969
	Z5	**3,926**	45.7823	15.6338	3.9540	0.6758	0.0083	0.0913
	Z6	**6,146**	32.7061	20.9761	4.5800	0.5917	0.0043	0.0656
	Z7	**985**	66.7899	56.3000	7.5033	0.3060	0.0160	0.1264
	Z8	**872**	46.4088	19.1136	4.3719	0.3809	0.0101	0.1003
	Z9	**11,422**	23.7836	45.7837	6.7664	0.3587	0.0087	0.0935
PCO	**Z1**	**18**	58.2077	10.6813	3.2682	0.9221	0.0002	0.0149
	Z2	**83**	51.2139	5.1880	2.2777	0.9202	0.0003	0.0171
	Z3	**0**	---	---	---	---	---	---
	Z4	**2,089**	60.2583	19.5085	4.4168	0.6661	0.0101	0.1007
	Z5	**12,693**	45.9560	15.9949	3.9994	0.6816	0.0082	0.0906
	Z6	**17,132**	33.0720	20.1777	4.4920	0.5952	0.0045	0.0667
	Z7	**2,107**	65.7038	53.9078	7.3422	0.3205	0.0144	0.1199
	Z8	**2,578**	46.3801	18.4695	4.2976	0.3815	0.0104	0.1021
	Z9	**25,184**	24.4796	46.3376	6.8072	0.3673	0.0083	0.0910

4.4. Temporal Coherence Analysis

The temporal coherence of each CS list was investigated. Each CS point that is output from the temporal coherence analysis, as described in Chapter 3, is evaluated. Statistically the temporal coherence mean values are similar for each CS list as shown in Table 10. The largest difference observed is the standard deviation of the cross-polarization HV list with the few CS points. Temporal coherence histograms are shown in Figure 23. Inspection of temporal coherence histograms show a bunching of points near the highest coherence value of one. The PCO CS list has the most CS points. The PCO list maintains high temporal coherence with a noticeable peak in values near a temporal coherence value of one. Thus, temporal coherence is maintained at high values with the full polarization PCO CS list. In other words, full polarization PCO increased CS quantity significantly compared to single polarizations while maintaining high temporal coherence, which may be one measure of CS quality.

The temporal coherence points from the analysis are superimposed on a power image for each polarization, HH, HV, VV, and full polarization PCO. As discussed earlier in this chapter, the overlap amongst the single polarization CS points is low. The full polarization CS list has denser coverage over the study. In addition, higher values of temporal coherence (i.e. yellow to cyan) are clearly seen in the town area as well as the more sparsely populated and vegetative areas. A few places around the corners of the images still have few CS even for full polarization PCO. Thus, full polarization is shown to have high temporal coherence in a denser fashion for the study site while a few areas within the study site remain challenging.

Table 10. Temporal Coherence Statistics by polarization CS list is tabulated. Statistics of mean, variance, and standard deviation are represented by μ, σ^2, and σ, respectively.

CS List Polarization	Number of Points	Temporal Coherence Statistics		
		μ	σ^2	σ
HH	20851	0.8800	0.0345	0.1858
HV	14338	0.8145	0.0499	0.2234
VV	20533	0.8773	0.0356	0.1887
PCO	61515	0.8373	0.0342	0.1848

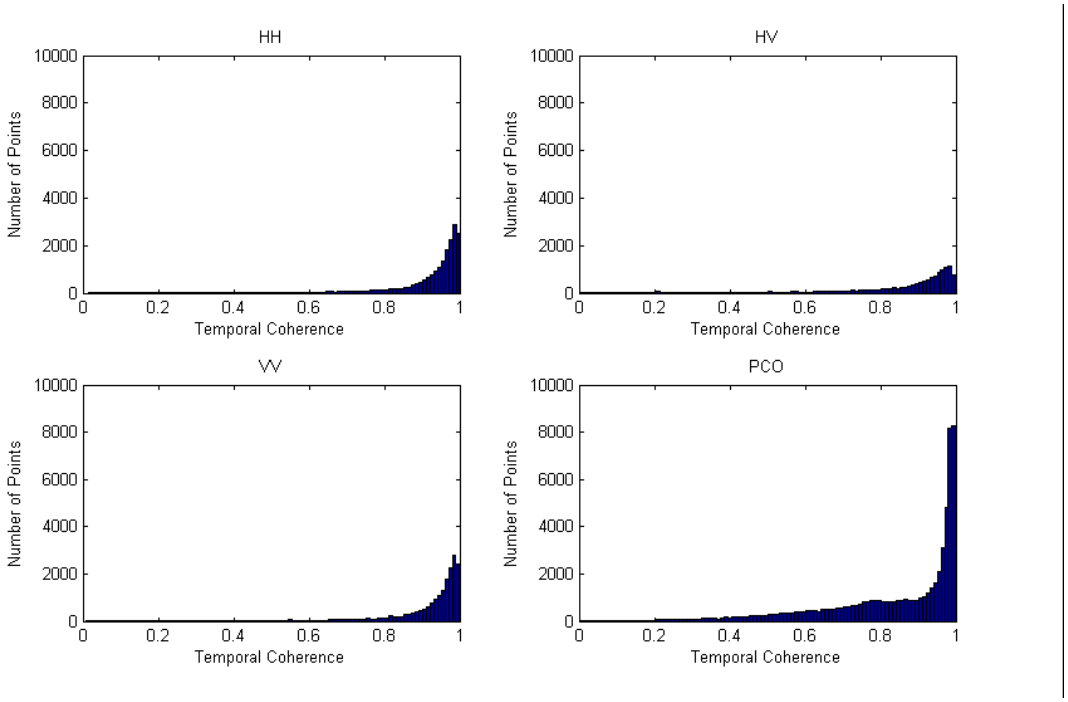


Figure 23. Temporal coherence CS list distribution histograms are presented by polarization. HH CS histogram is shown upper left. HV CS histogram is shown upper right. VV CS histogram is shown lower left. Full polarization PCO CS histogram is shown lower right.

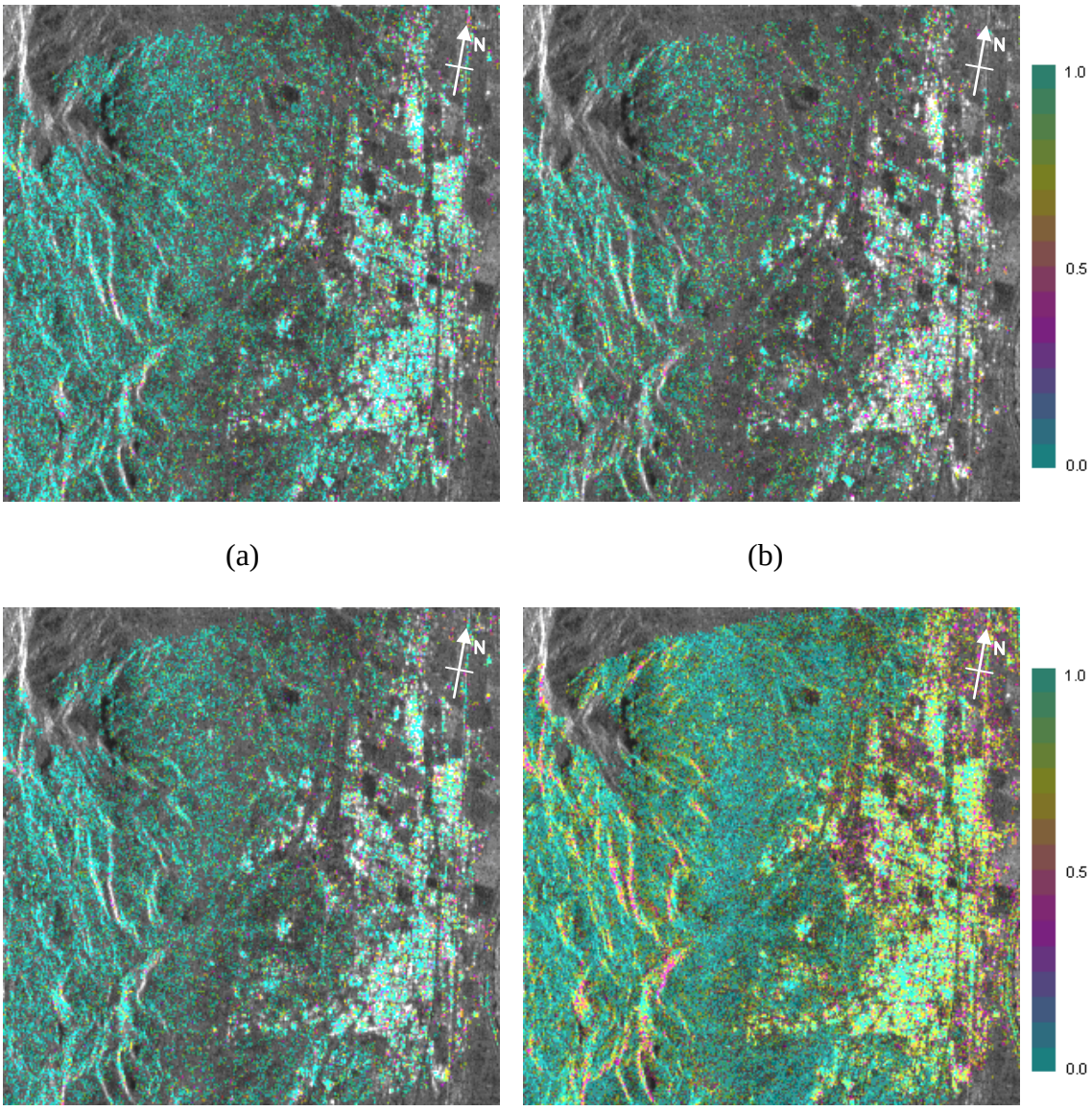


Figure 24. Images show temporal coherence map for each CS list: a) HH, b) HV, c) VV, and d) full polarization PCO.

4.5. Summary

Full polarization images were analyzed to study their potential for CS identification in InSAR Interferometric point target analysis and time-series analyses. Since InSAR provides a means of imaging small deformations of the Earth's surface (e.g. millimeter scale), PolInSAR processing techniques have the potential to improve detection of point targets or coherent scatterers that are characterized by a point-like scattering behavior.

In this study, full polarization techniques provided a significant increase (>250%) in CSs compared to a non full polarization. Using polarimetric decomposition, the CS scattering mechanism behavior was found consistent amongst the CS lists. Using temporal analysis, the temporal behavior of the CSs was explored. Full polarization PCO was found to increase the CS quantity significantly compared to single polarizations while maintaining high temporal coherence.

This significant increase in full polarization CSs may increase the overall InSAR processing quality. For sites that challenge (e.g. vegetated areas and other low coherence environments) the single polarization InSAR processing, the full polarization increase in CSs may be essential for processing. A larger number of images and quantity of geographic sites then were available for this study is needed to further quantify.

Chapter 5. Conclusion and Future Work

5.1. Conclusion

For the study site of Socorro, New Mexico, full polarization techniques provided a significant increase (i.e. more than doubled) in CSs compared to a non full polarization. Thus, this significant increase in full polarization CSs has the potential to increase the overall InSAR processing quality, which may enhance the ability of imaging small deformations of the Earth's surface (millimeter scale) using InSAR.

Comparison of the full polarization PCO result to HH, HV, and VV polarizations was conducted. Overlap amongst the single polarization CS lists was small (<8%). In contrast, the vast majority (>97%) of the HH, HV, and VV polarization CSs overlap with full polarization PCO CS list. Thus, the full polarization CS list contains more than 2.5 times the largest single polarization CSs from the HH single polarization.

For full polarization, the increase (>250%) in CSs has a number of potential benefits. This significant increase in full polarization CSs may increase the overall InSAR processing quality. For sites that challenge (e.g. vegetated areas and other low coherence environments) the single polarization InSAR processing, the full polarization increase in CSs may be essential for processing. A larger number of images and quantity of geographic sites then were available for this study is needed to further quantify.

For the study site, the CSs were dominated in each decomposition case. For Alpha-Entropy, zones 6 and 9 correspond to surface scattering mechanisms. HH, HV, VV, and PCO CS lists contain about 71.6%, 59.0%, 73.2%, and 68.4% CS points

classified within the Alpha-Entropy zones 6 and 9. The CS lists each corresponded to about 39% of Pauli odd bounce parameter K1, which approximately corresponds to a physical interpretation as a combination of surface, sphere, corner reflector scattering. The study site is not geographically dominated with man-made structures. Thus, the decomposition results seem to align qualitatively well with the local geography in scattering mechanism.

The temporal coherences of the CS lists were examined for each polarization, HH, HV, VV, and full polarization PCO. The full polarization CS list has denser coverage over the study. In addition, higher values of temporal coherence are clearly seen in the town area as well as the more sparsely populated and vegetative areas. A few places around the corners of the images still have few CS even for full polarization PCO. Thus, full polarization is shown to have high temporal coherence in a denser fashion for the study site while a few areas within the study site remain challenging.

5.2. Future Work

A number of directions could be explored in future studies. Acquisition of additional full polarization data sets would facilitate additional investigations. For the Socorro, New Mexico site, five fully polarimetric images were acquired with an acquisition span of about 3 months. Additional acquisitions (e.g. 20+) and time span expansion (e.g. regular time sampling over years) are desired for each study site. With additional data stacks over multiple geographic sites, investigation of site dependencies and variations should provide insightful to InSAR ability of imaging small deformations of the Earth's surface (millimeter scale) under various scenarios.

This study utilized publicly available RADARSAT-2 C-Band (5.405 GHz) fully polarimetric data sets. If sensors with other frequencies and configurations become publicly available in the future, additional study and comparisons could be explored as well. Studies of ground based measurements have value to advancing the full polarization processing and better understanding the data of space based systems. For instance, ground based radar that is being developed at the University of Missouri could provide a test bed for such an effort.

Chapter 6. References

Balanis, C. A. (1989). *Advanced Engineering Electromagnetics.* New York: John Wiley & Sons.

Bamler, R. and P. Hartl, (1998). "Synthetic aperture radar interferometry," *Inverse Problems*, Vol. 14, No. R1.

Boerner, W. M., W. L. Yan, A. Q. Xi, and Y. Yamaguchi, (1991). "On the basic principles of radar polarimetry: The target characteristic polarization state theory of Kennaugh, Huynen's polarization fork concept, and its extension to the partially polarized case," *Proceedings of the IEEE* , Vol. 79, No. 10, pp. 1538-1550.

Cloude, S. R. and E. Pottier, (1996). "A review of target decomposition theorems in radar polarimetry," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 34, No. 2, pp. 498-518.

Cloude, S. R., and E. Pottier. (1997). "An entropy based classification scheme for land applications of polarimetric SAR." *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 35, No. 1, pp. 68-78.

Cloude, S. R. and K. P. Papathanassiou, (1998). "Polarimetric SAR interferometry," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 36, No. 5, pp. 1551-1565.

Elachi, C., and J. van Zyl, (2006). *Introduction to the Physics and Techniques of Remote Sensing*, Second Edition. Hoboken, New Jersey: Wiley-Interscience.

Ferretti, A., C. Prati, and F. Rocca, (2001). "Permanent scatterers in SAR interferometry," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 39, No. 1, pp. 8-20.

Freeman, A., and S. L. Durden, (1998). "A three-component scattering model for polarimetric SAR data," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 36, No. 3, pp. 963-973.

Gamma Remote Sensing AG. (2009). GAMMA SAR. [Software]. Bern, Switzerland: Gamma Remote Sensing AG.

Google, Inc. (2010, March 27). Google Earth (Version 5.1.3534.0411) [Software]. Available from http://earth.google.com/download-earth.html.

Jensen, J. R., (2007). *Introductory Digital Image Processing: A Remote Sensing Perspective*. Third Edition. Upper Saddle River, NJ: Pearson Prentice Hall.

Lee, J. S., and E. Pottier, (2009). *Polarimetric Radar Imaging: From Basics to Applications.* Boca Raton, Florida: CRC Press.

Livingstone, C. E., I. Sikaneta, C. Gierull, S. Chiu, and P. Beaulne, (2005). "RADARSAT-2 system and mode description," *Integration of Space-Based Assets within Full Spectrum Observations*, Meeting Proceedings RTO-MP_SCI-150, Paper 15, pp.15-1 – 15-22.

MacDonald, Dettwiler and Associates Ltd. (2009, November 2). *RADARSAT-2 Product Description* (RN-SP-52-1238). Richmond, B.C., Canada: MacDonald, Dettwiler and Associates Ltd.

MacDonald, Dettwiler and Associates Ltd. (2008, March 14). *RADARSAT-2 Product Format Definition* (RN-RP-51-2713). Richmond, B.C., Canada: MacDonald, Dettwiler and Associates Ltd.

Richards, M. A. (2005). *Fundamentals of Radar Signal Processing.* New York, NY: McGraw-Hill.

Rosen, P. A., S. Hensley, I. R. Joughin, F. K. Li, S. N. Madsen, E. Rodriguez, and R. M. Goldstein, (2000). "Synthetic aperture radar interferometry," *Proceedings of the IEEE*, Vol. 88, No. 3, pp. 333-382.

Sarmap SA. (2008). ENVI SARscape (Version 4.1.001). [Software]. Lugano, Switerland: Sarmap SA.

The MathWorks, Inc. (2007, January 29). MATLAB (Version 7.4.0.287) [Software]. Natick, Massachusetts: The MathWorks Inc.

Ulaby, F. T. (2004). *Fundamentals of Applied Electromagnetics*, 2004 Media Edition. Upper Saddle River, New Jersey: Pearson Prentice Hall.

U.S. Geological Survey (2006, August). *National Elevation Dataset*. Retrieved on February 8, 2010 from http://ned.usgs.gov/.

Zebker, H. A., and J. van Zyl, (1991). "Imaging radar polarimetry: A review," *Proceedings of the IEEE*, Vol. 79, No. 11, pp. 1583-1606.

Chapter 7. APPENDICES

7.1. Appendix A: Point Lists Locations

Table 11. Number of point targets in the initial and final point scatterer lists.

Polarization	Initial Point List Size	Final Point List Size	Points Pruned	Percent Pruned
HH	26,466	23,951	2515	9.50
HV	22,095	19,273	2822	12.77
VV	26,321	24,124	2197	8.35
PCO	68,546	61,884	6662	9.72

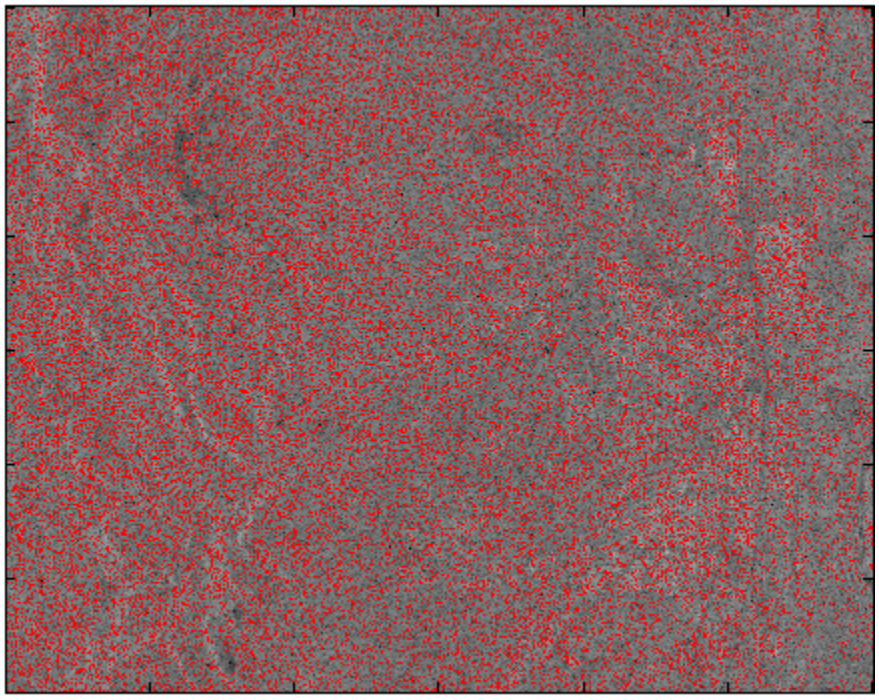


Figure 25. Initial HH point list locations. Points are shown in red.

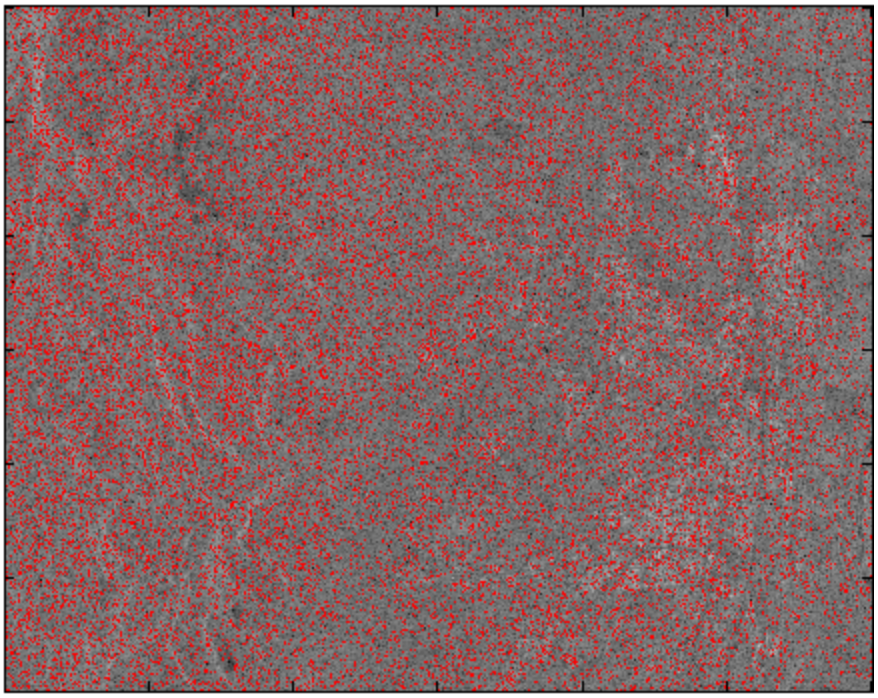


Figure 26. Initial HV point list locations. Points are shown in red.

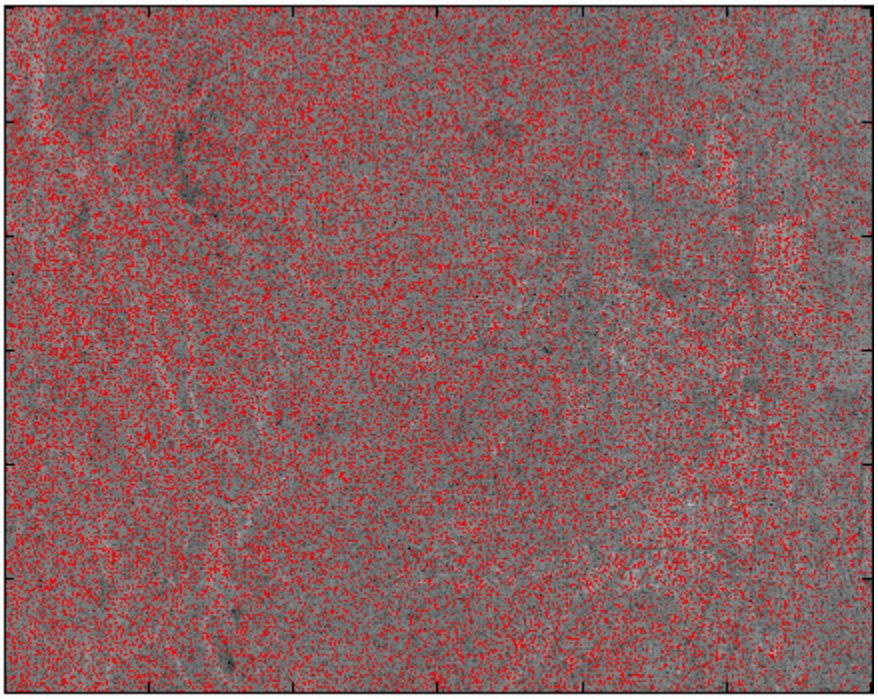


Figure 27. Initial VV point list locations. Points are shown in red.

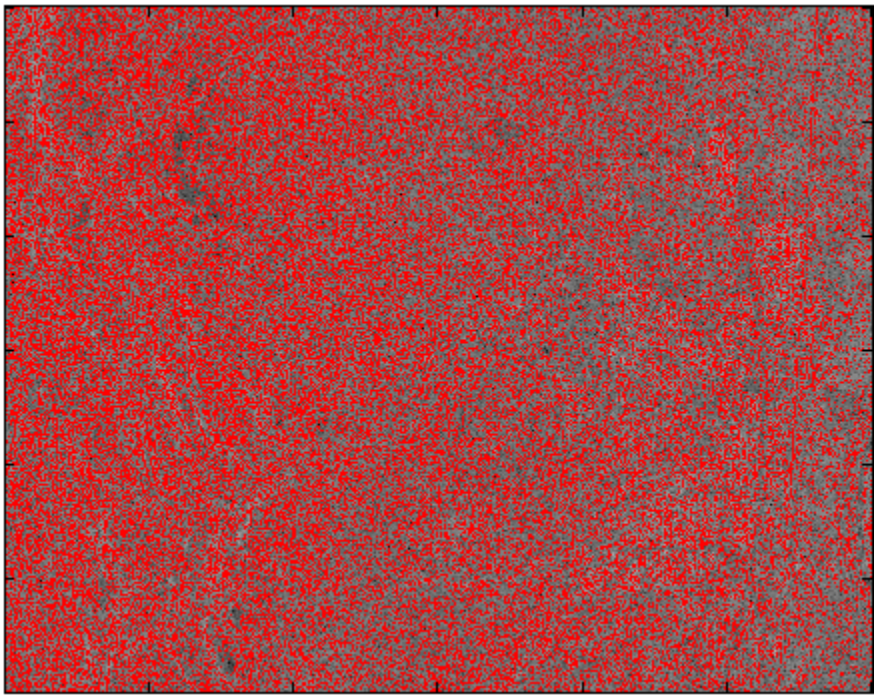


Figure 28. Initial PCO point list locations. Points are shown in red.

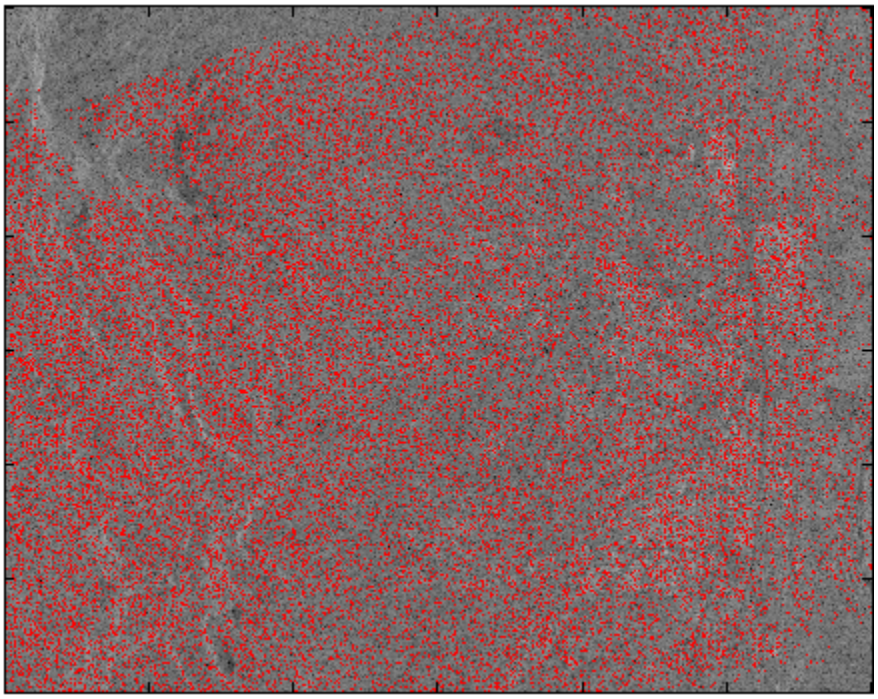


Figure 29. Final HH point list locations. Points are shown in red.

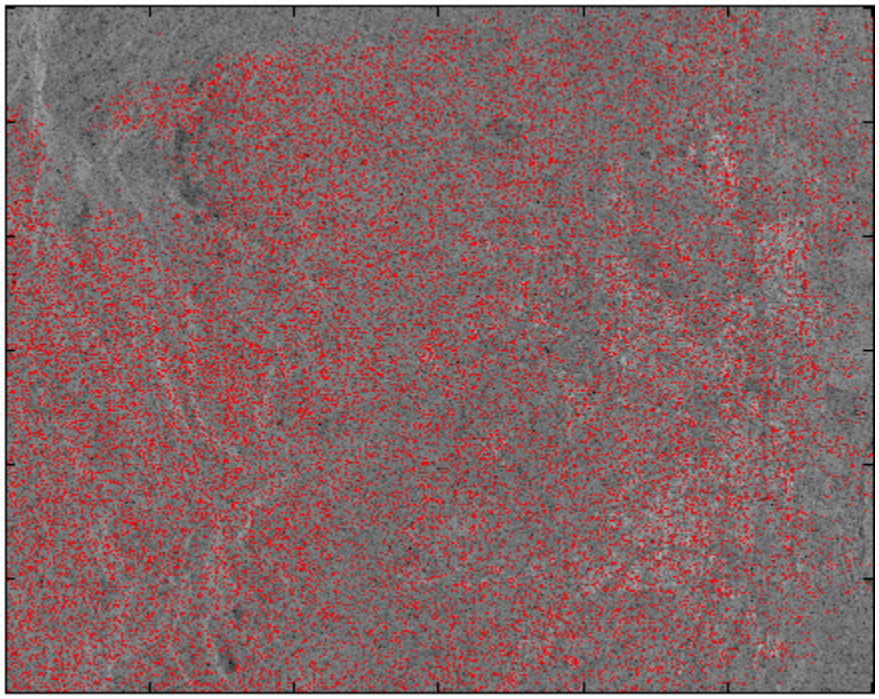


Figure 30. Final HV point list locations. Points are shown in red.

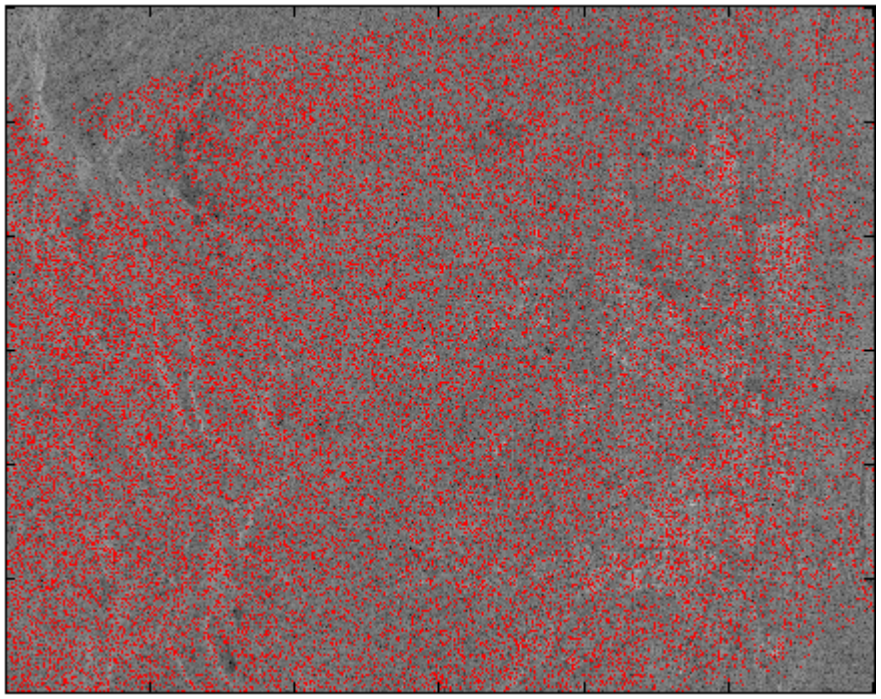


Figure 31. Final VV point list locations. Points are shown in red.

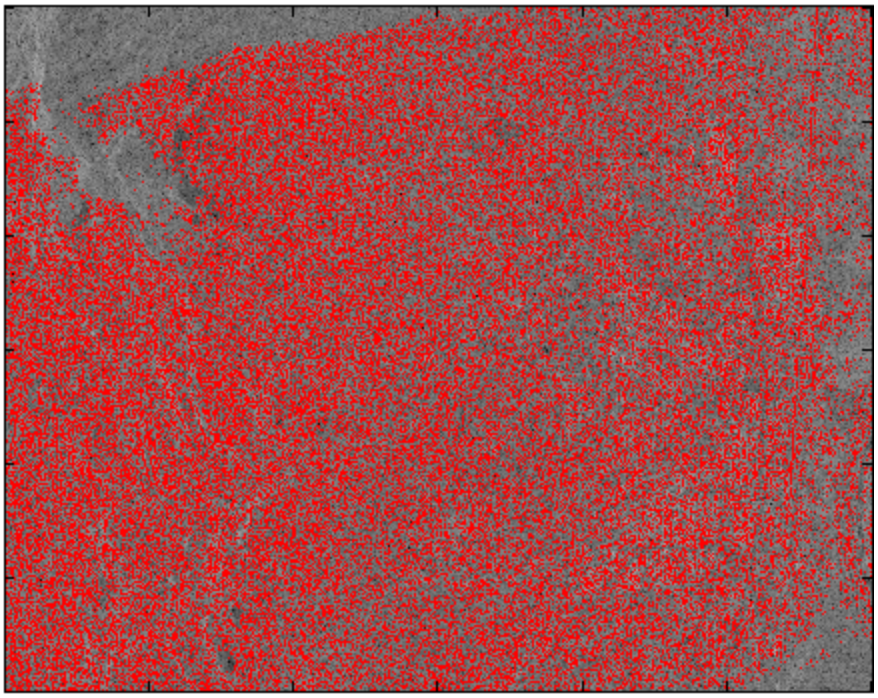


Figure 32. Final PCO point list locations. Points are shown in red.

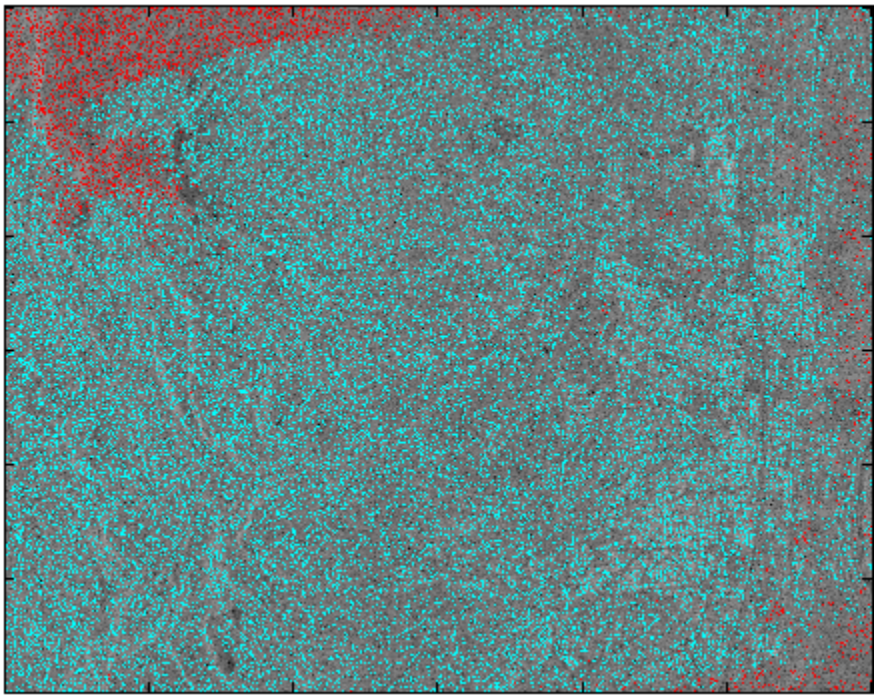


Figure 33. Comparison of initial and final HH point lists. Final point locations are shown in blue and pruned points are shown in red.

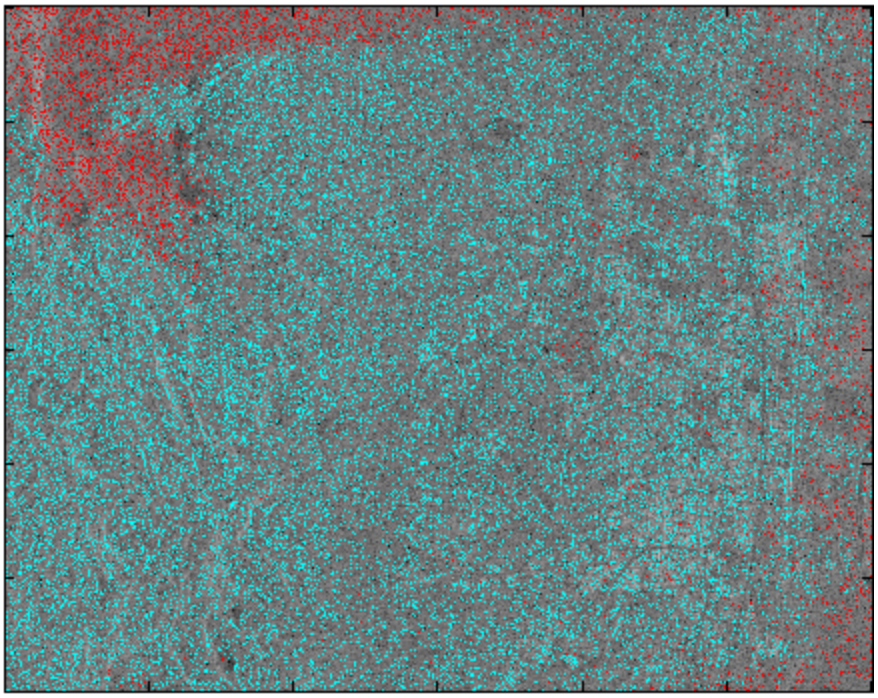


Figure 34. Comparison of initial and final HV point lists. Final point locations are shown in blue and pruned points are shown in red.

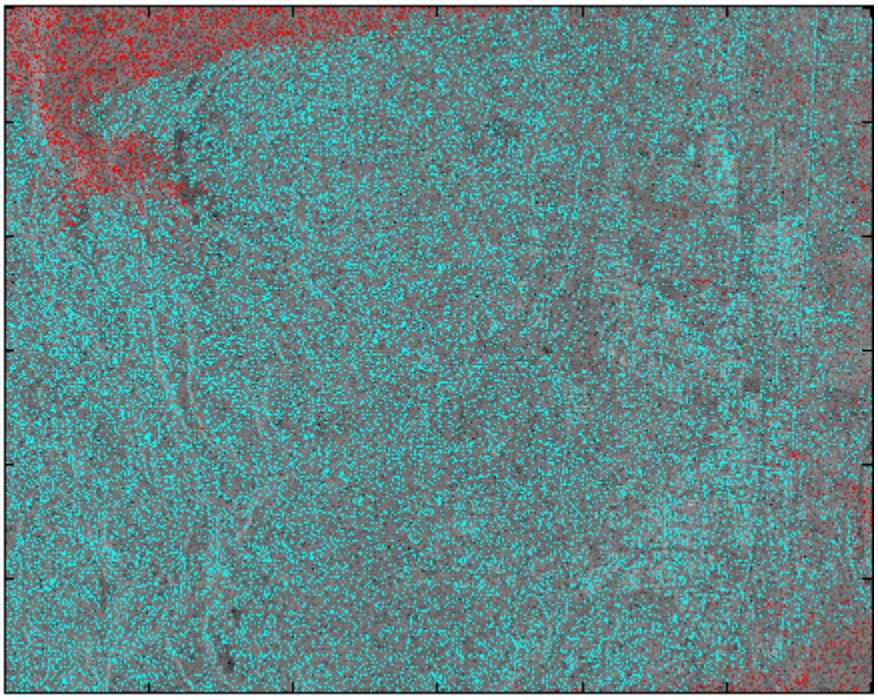


Figure 35. Comparison of initial and final VV point lists. Final point locations are shown in blue and pruned points are shown in red.

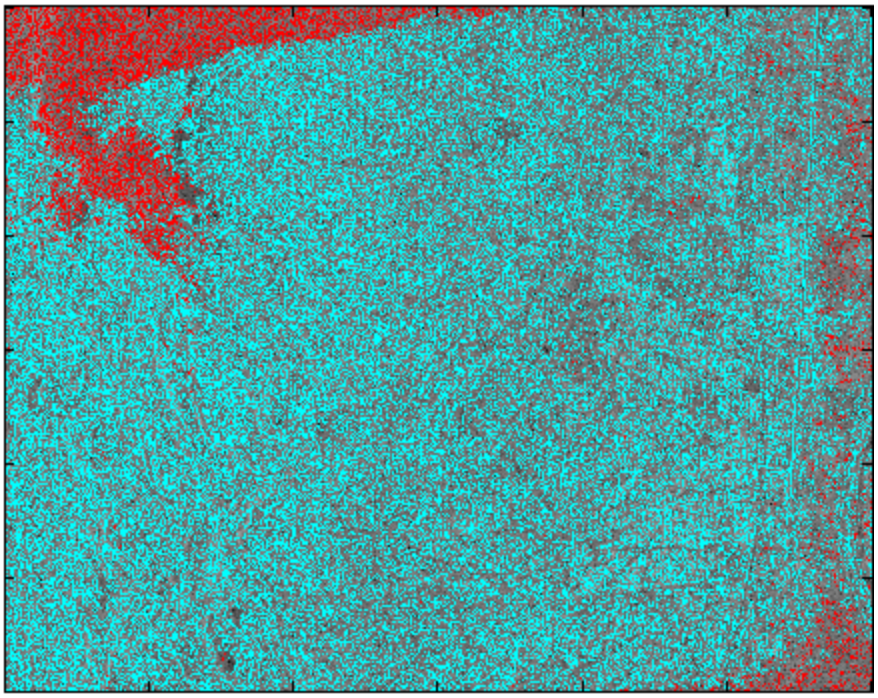


Figure 36. Comparison of initial and final PCO point lists. Final point locations are shown in blue and pruned points are shown in red.

Table 12. Point target density for final single-pol point lists.

Number of occurances	Number of Points	Percent of Total
1	56242	91.23
2	5106	8.28
3	298	0.48

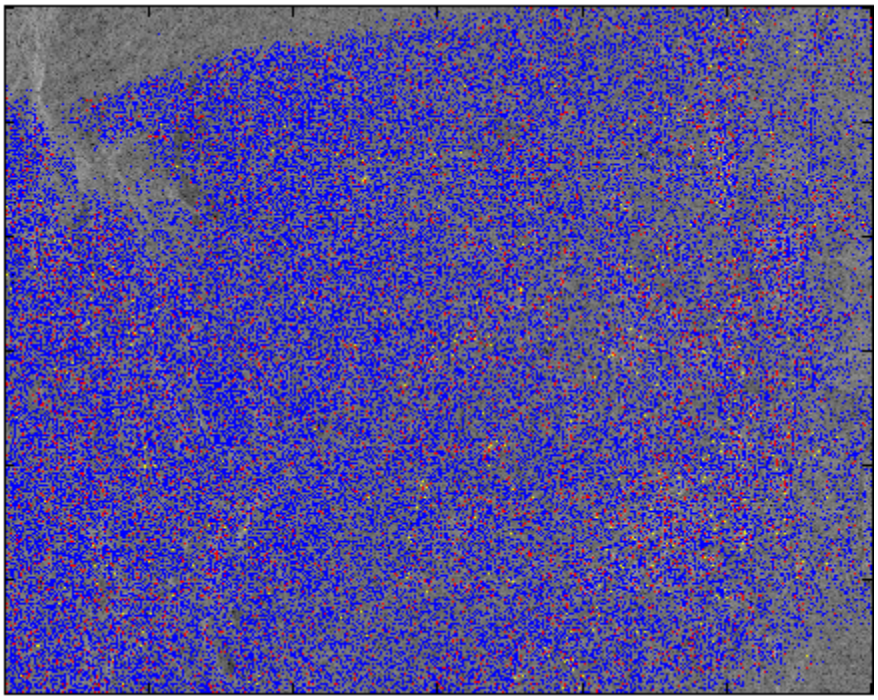


Figure 37. Point target density among single-pol point lists. Number of occurrences per location is indicated by color: Blue = 1, Red =2, Yellow =3.

7.2. Appendix B: Polarimetric Coherence Optimization Analysis

Table 13. Common point targets between single-pol and quad-pol final point scatterer lists.

Point List	Final Point List Size	Points in Common	Percent of Single-Pol List	Percent of Quad-Pol List
HH	23,951	23,429	97.82	37.86
HV	19,273	18,910	98.12	30.56
VV	24,124	23,421	97.09	37.85

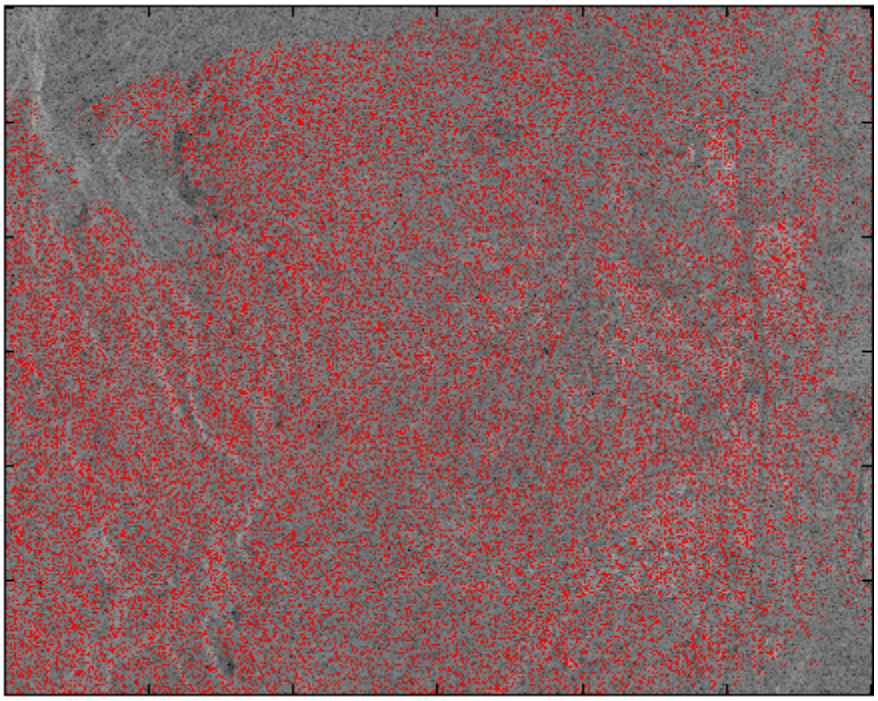


Figure 38. Common point target locations between PCO and HH point lists. Points are shown in red.

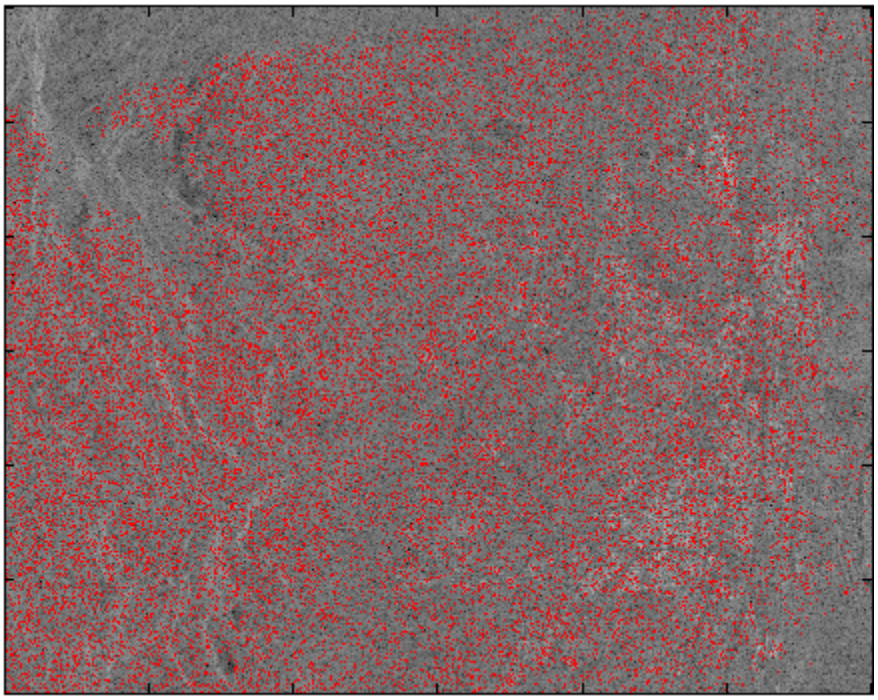


Figure 39. Common point target locations between PCO and HV point lists. Points are shown in red.

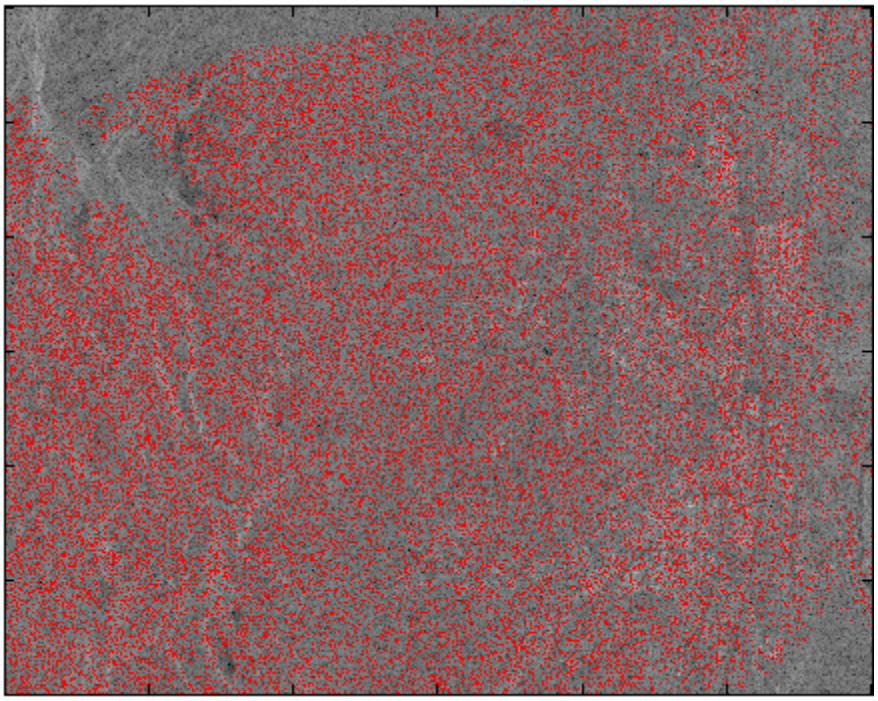


Figure 40. Common point target locations between PCO and VV point lists. Points are shown in red.

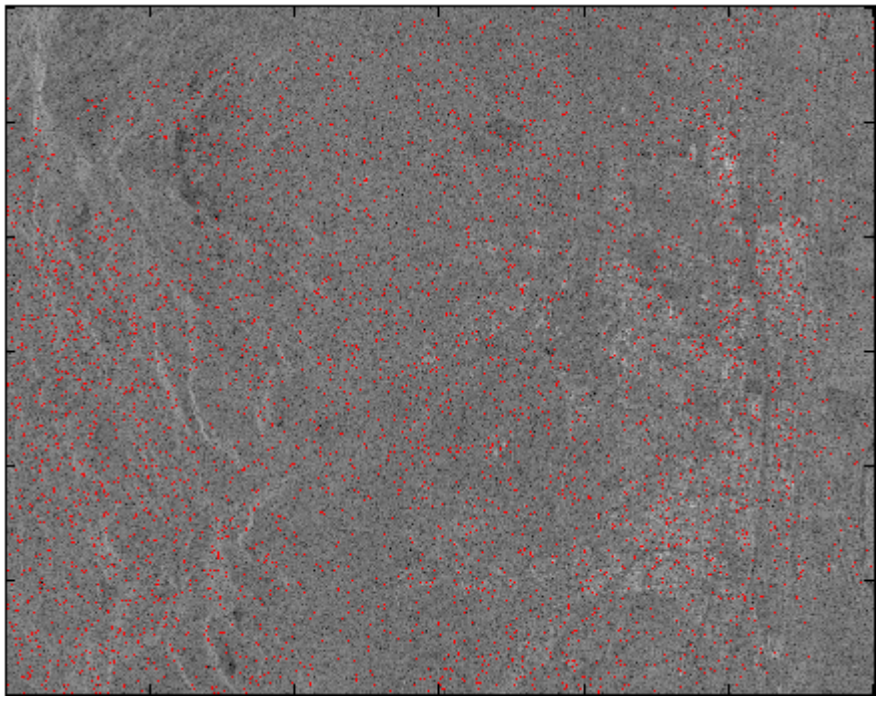


Figure 41. Common point target locations between HH and VV point lists. Points are shown in red.

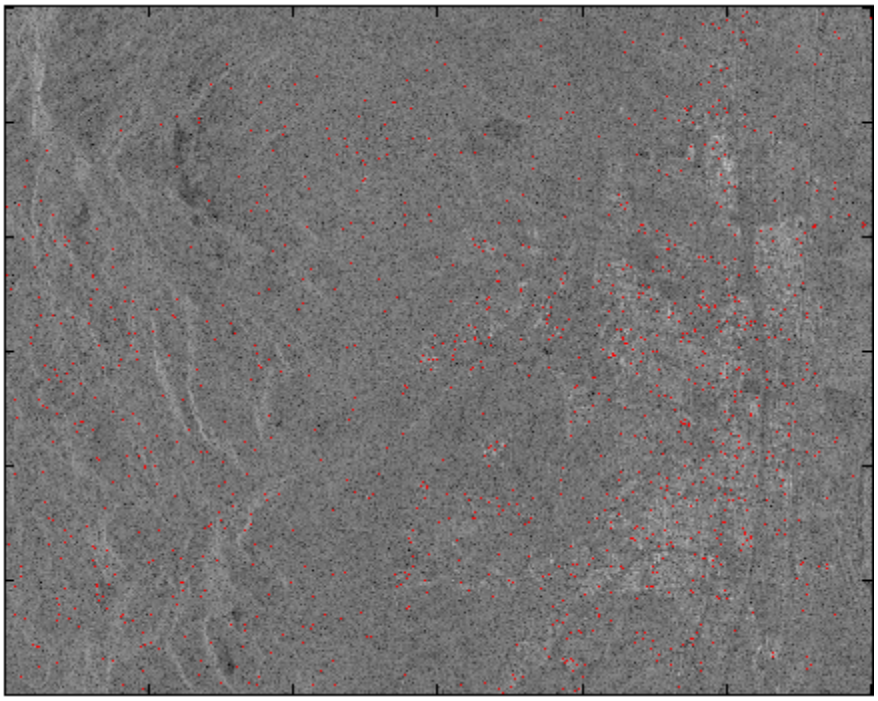


Figure 42. Common point target locations between HH and HV point lists. Points are shown in red.

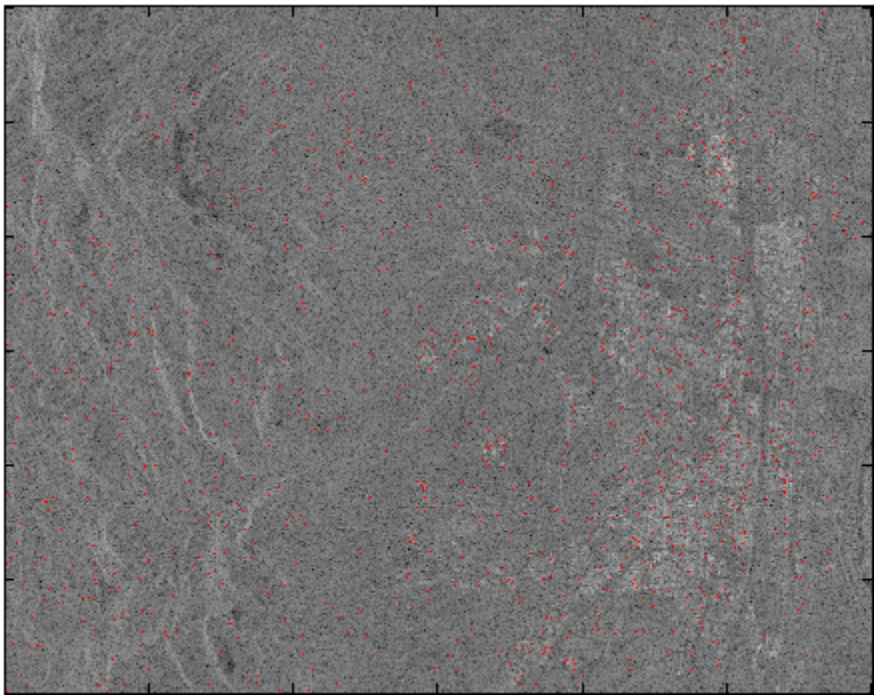


Figure 43. Common point target locations between VV and HV point lists. Points are shown in red.

Table 14. Temporal coherence statistics by polarization and scatterer type.

Polarization	Number Points	Mean	Variance	Std. Deviation
HH	20578	0.8823	0.0344	0.1855
HV	14425	0.8169	0.0501	0.2239
VV	20615	0.8793	0.0359	0.1895
PCO	61515	0.8373	0.0342	0.1848

7.3. Appendix C: Initial Unwrapped Differential Interferograms

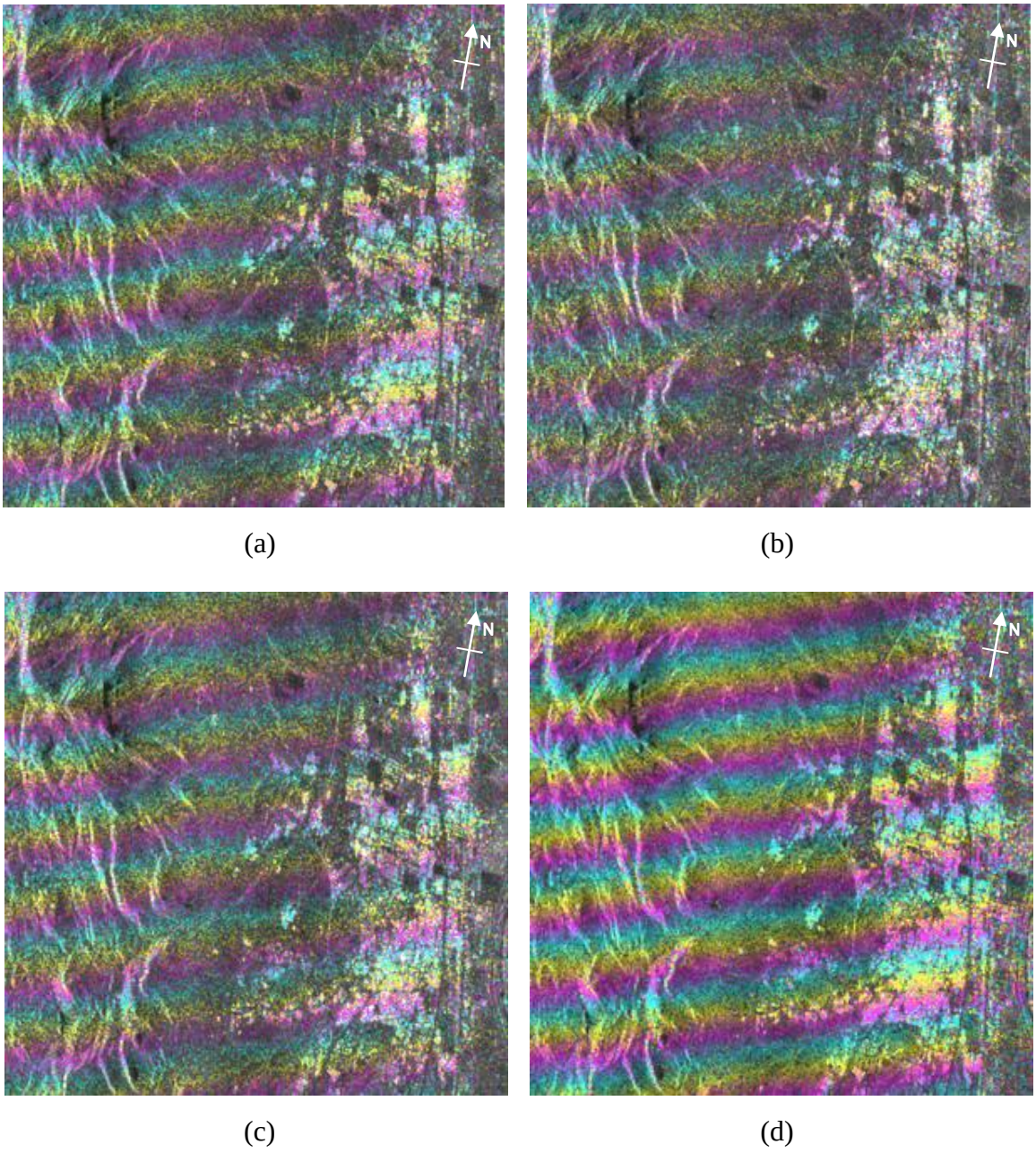
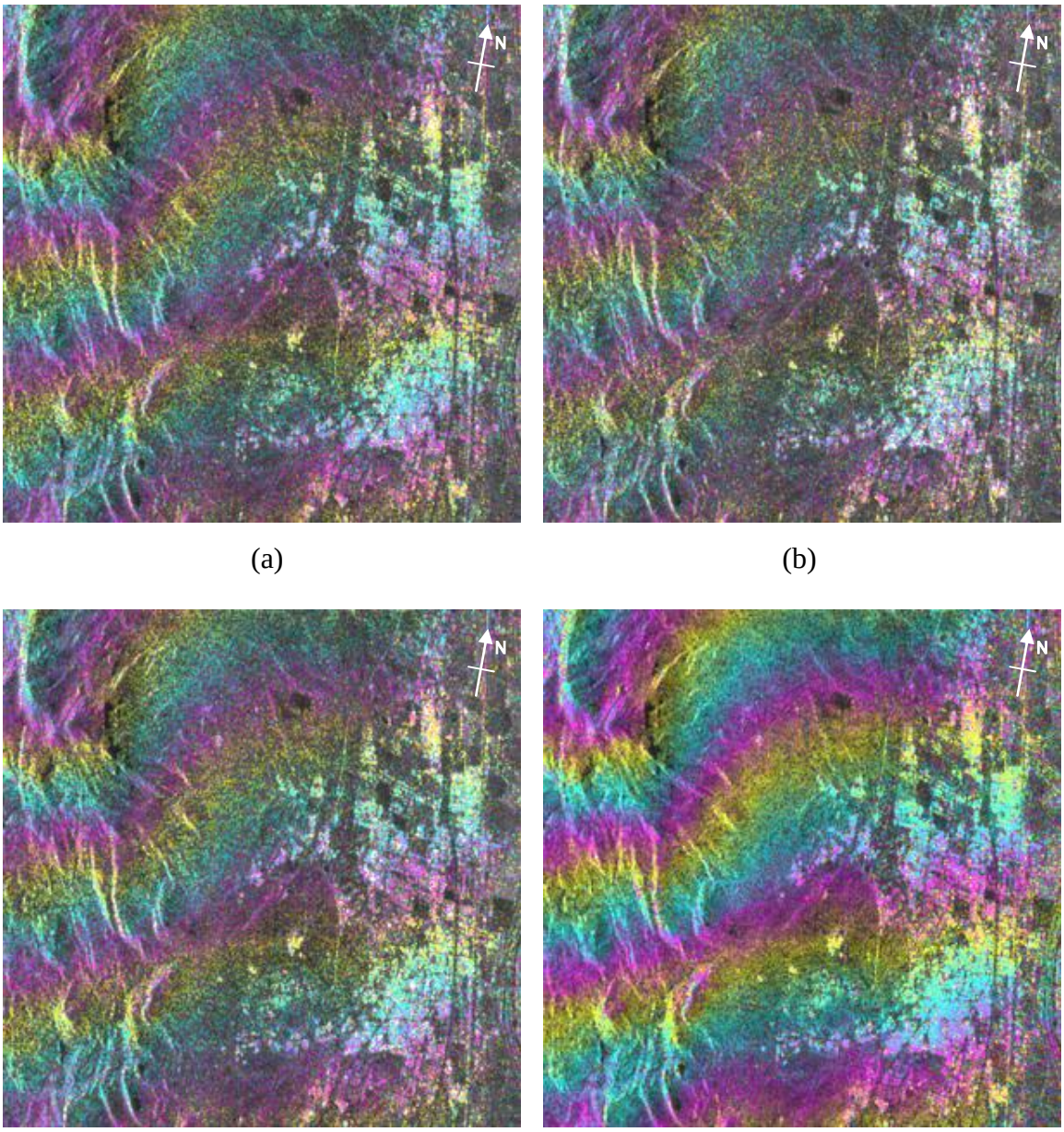


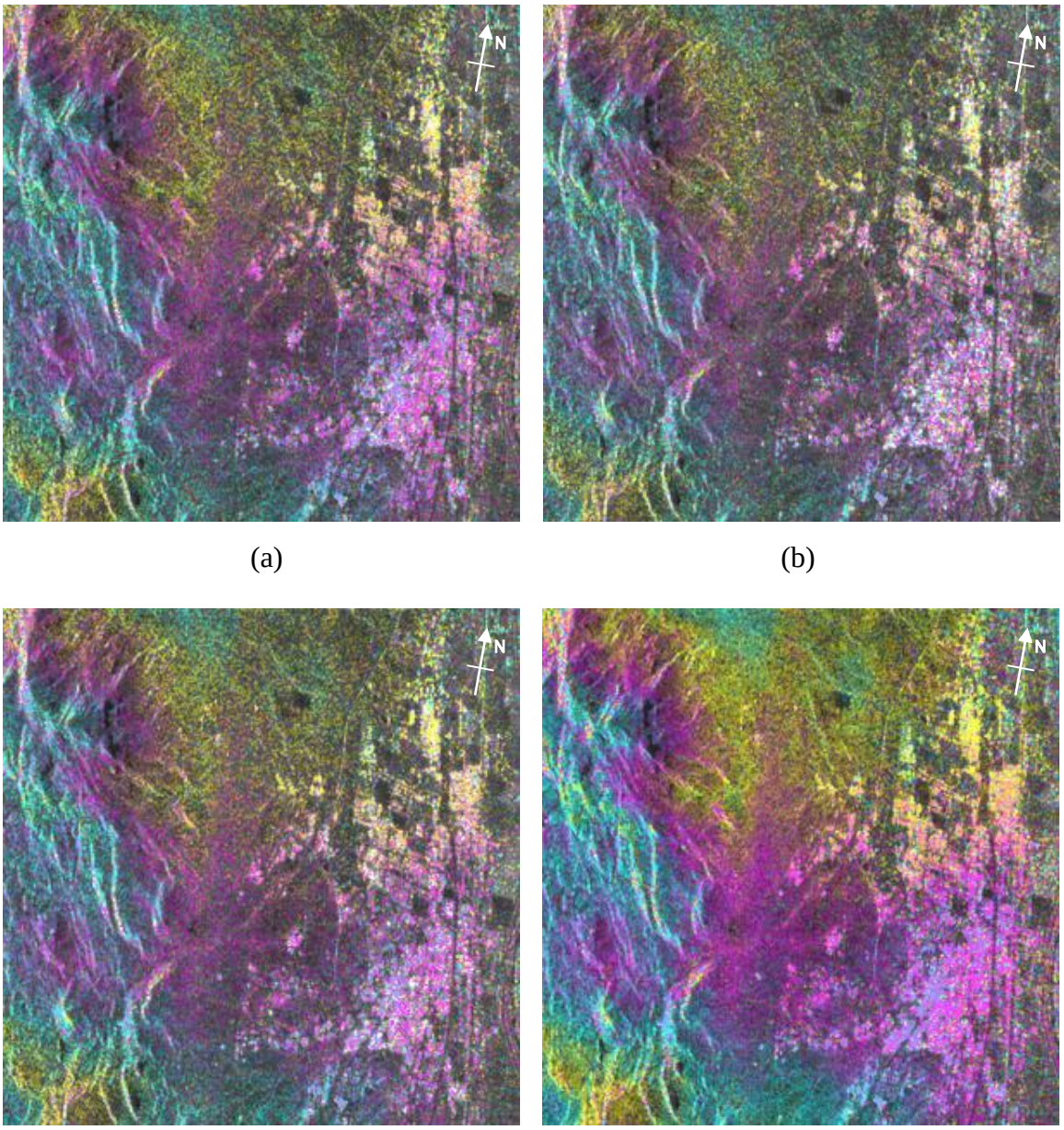
Figure 44. Initial differential interferograms, 20080802_20080615. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).



(a) (b)

(c) (d)

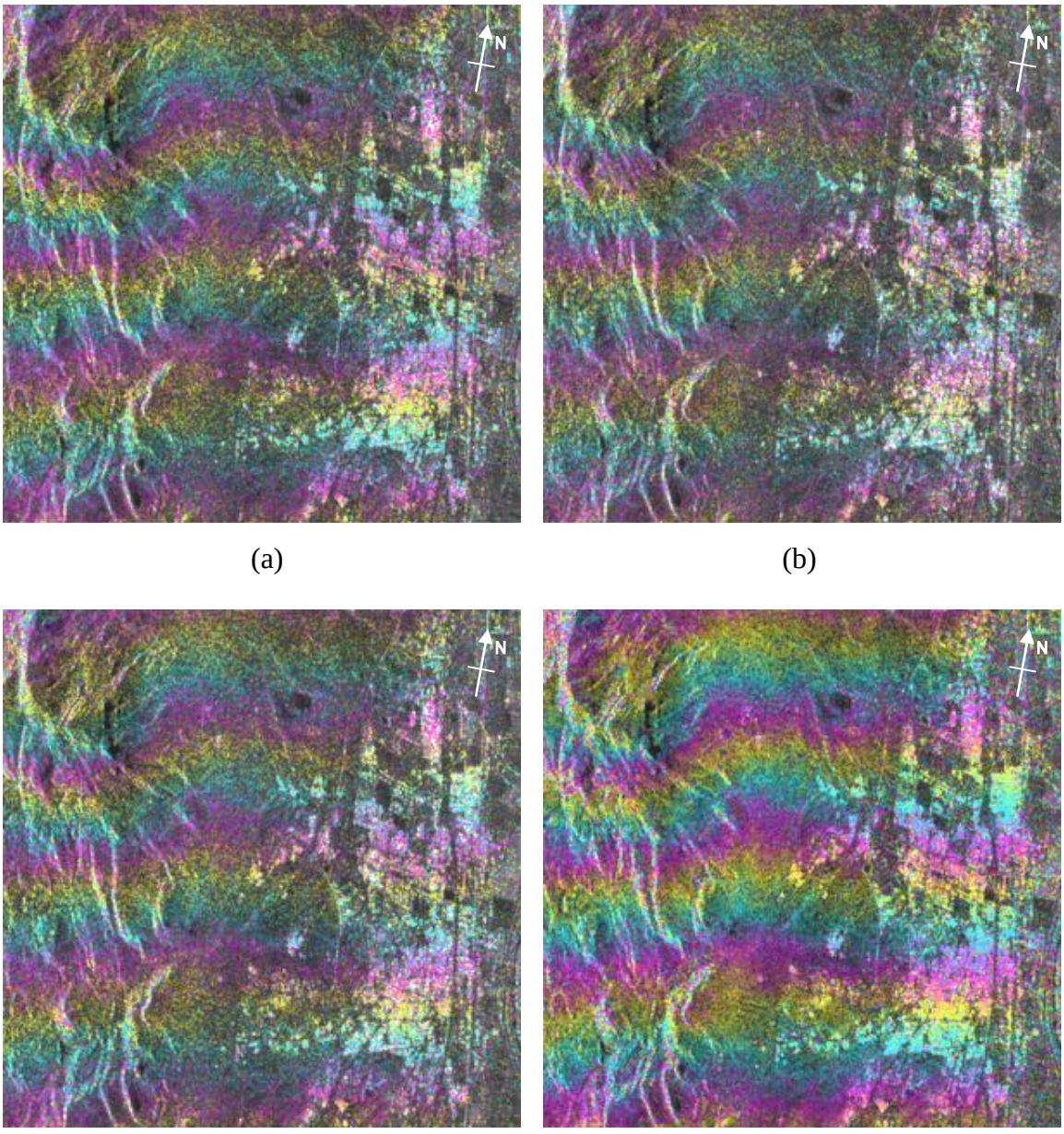
Figure 45. Initial differential interferograms, 20080802_20080709. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).



(a) (b)

(c) (d)

Figure 46. Initial differential interferograms, 20080802_20080826. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).



(a) (b)

(c) (d)

Figure 47. Initial differential interferograms, 20080802_20080919. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

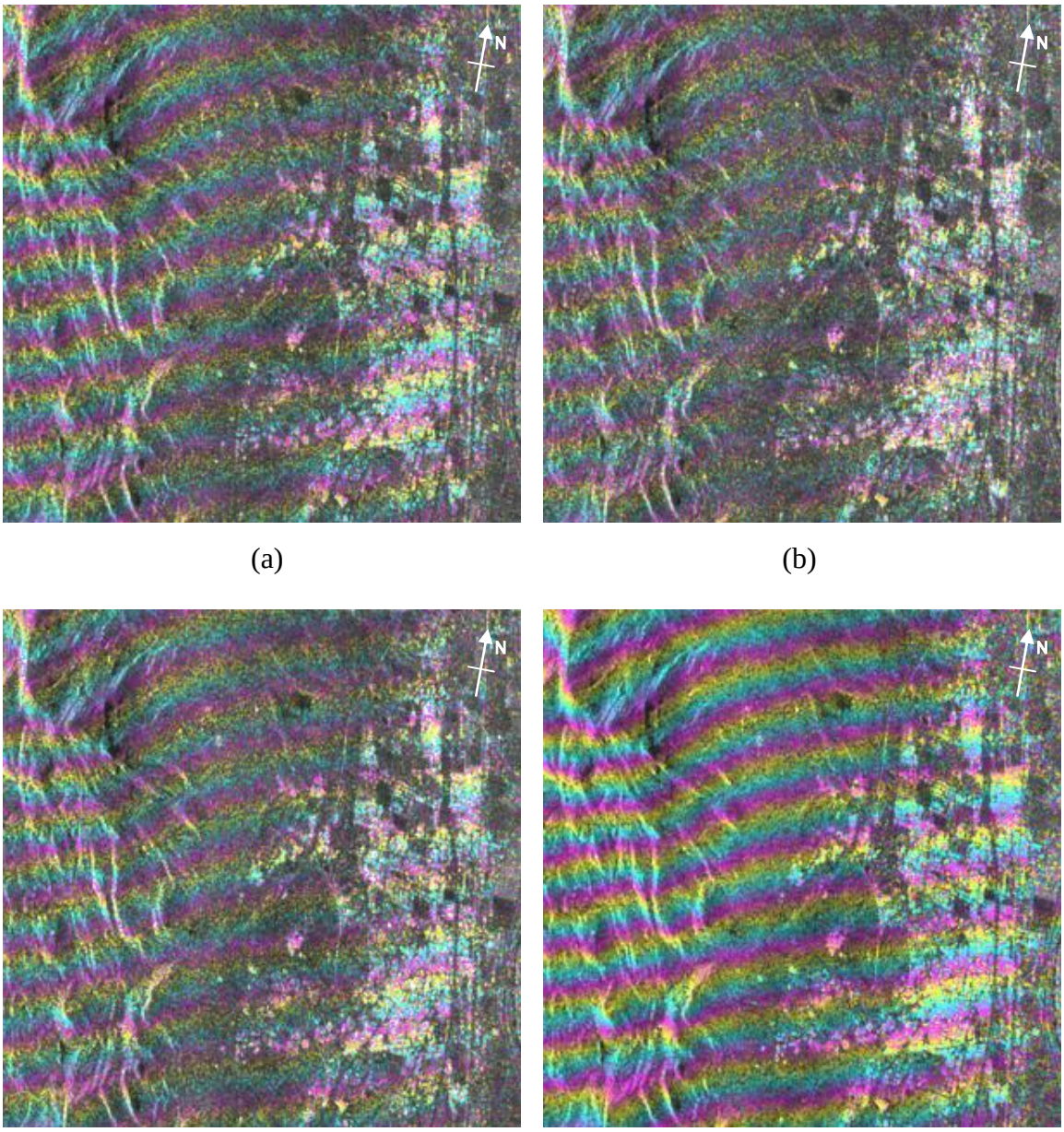


Figure 48. Initial differential interferograms, 20080615_20080709. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

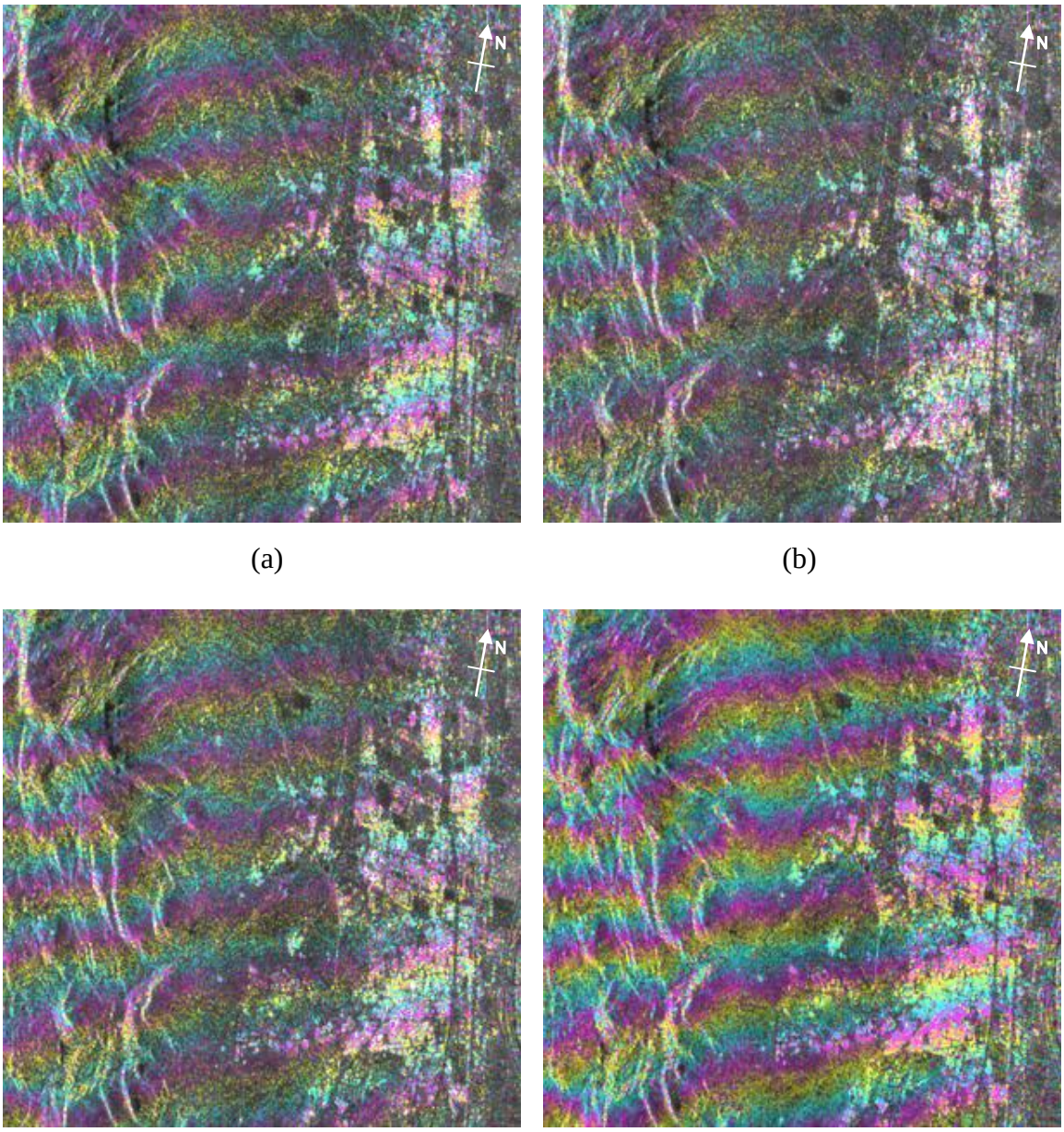


Figure 49. Initial differential interferograms, 20080615_20080826. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

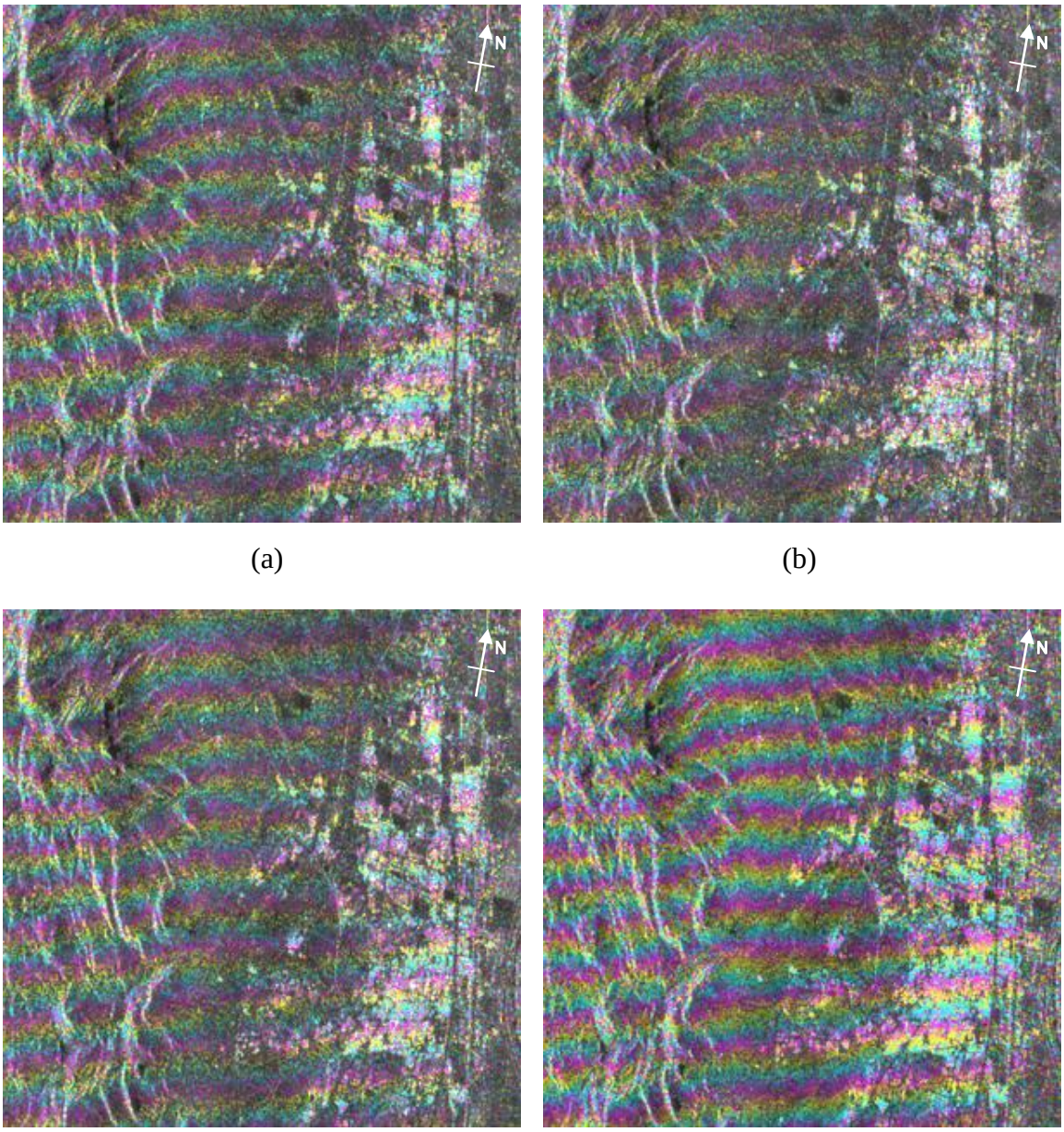


Figure 50. Initial differential interferograms, 20080615_20080919. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

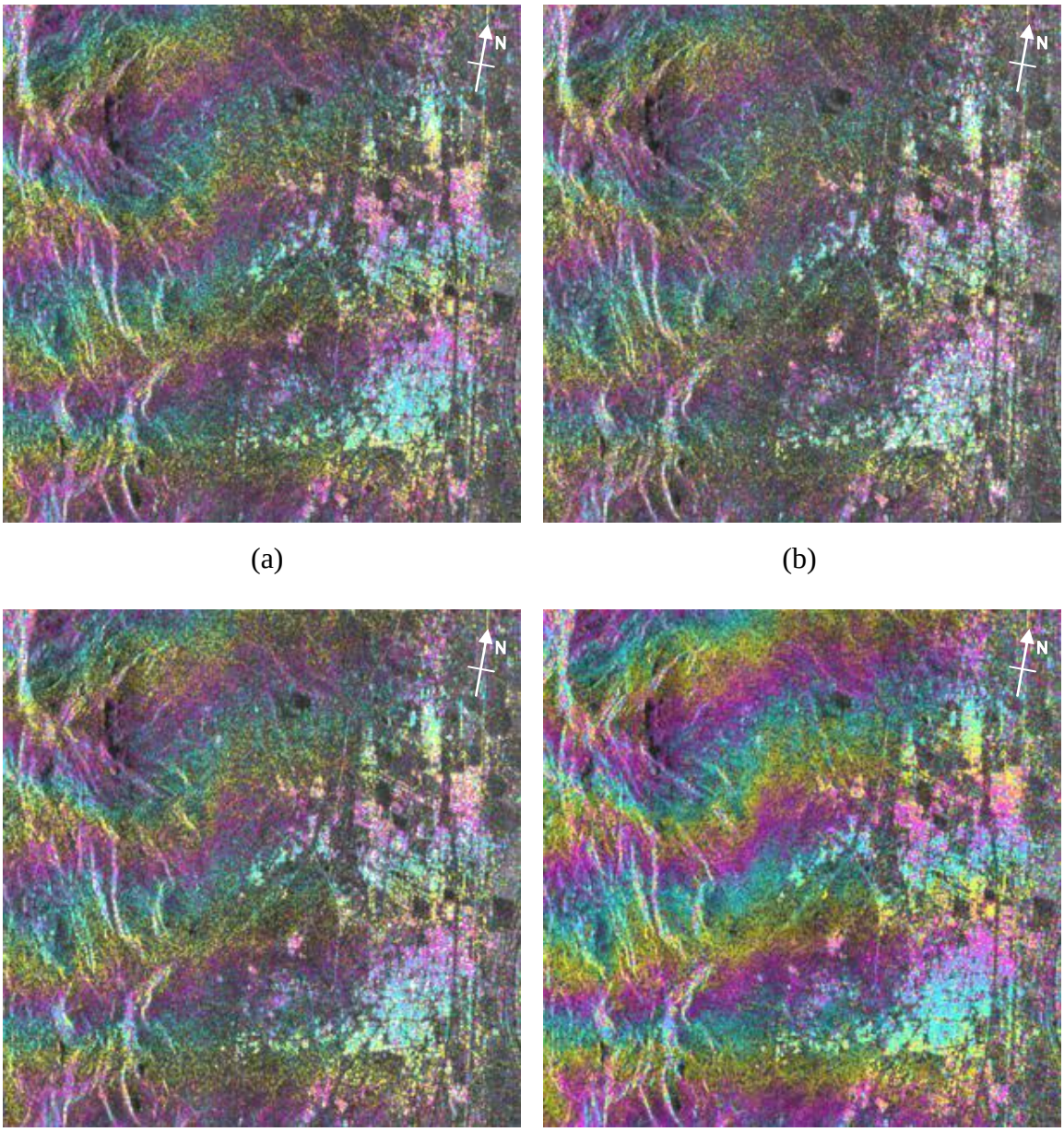
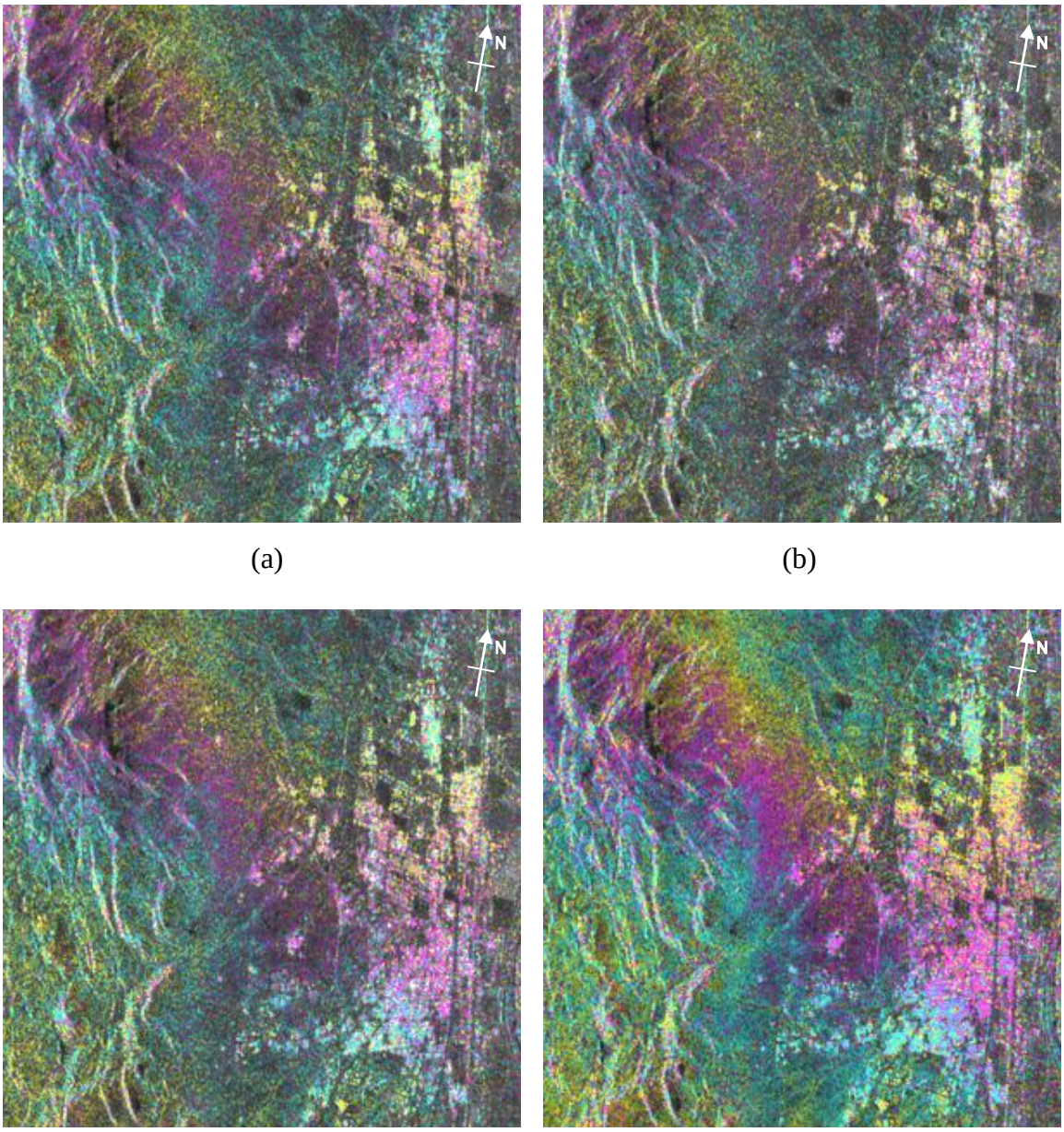


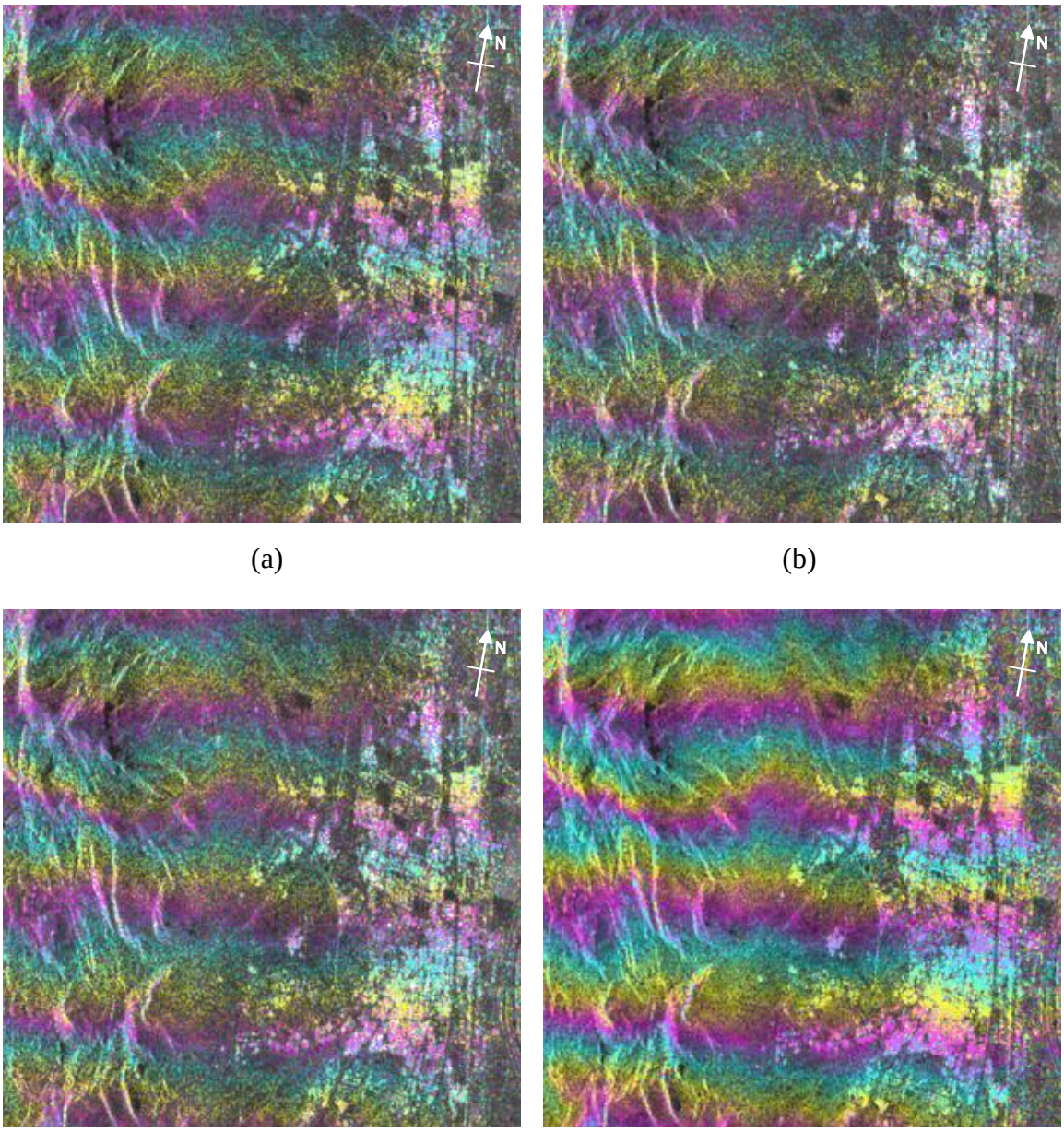
Figure 51. Initial differential interferograms, 20080709_20080826. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).



(a) (b)

(c) (d)

Figure 52. Initial differential interferograms, 20080709_20080919. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).



(a) (b)

(c) (d)

Figure 53. Initial differential interferograms, 20080826_20080919. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

7.4. Appendix D: Final Unwrapped Differential Interferograms

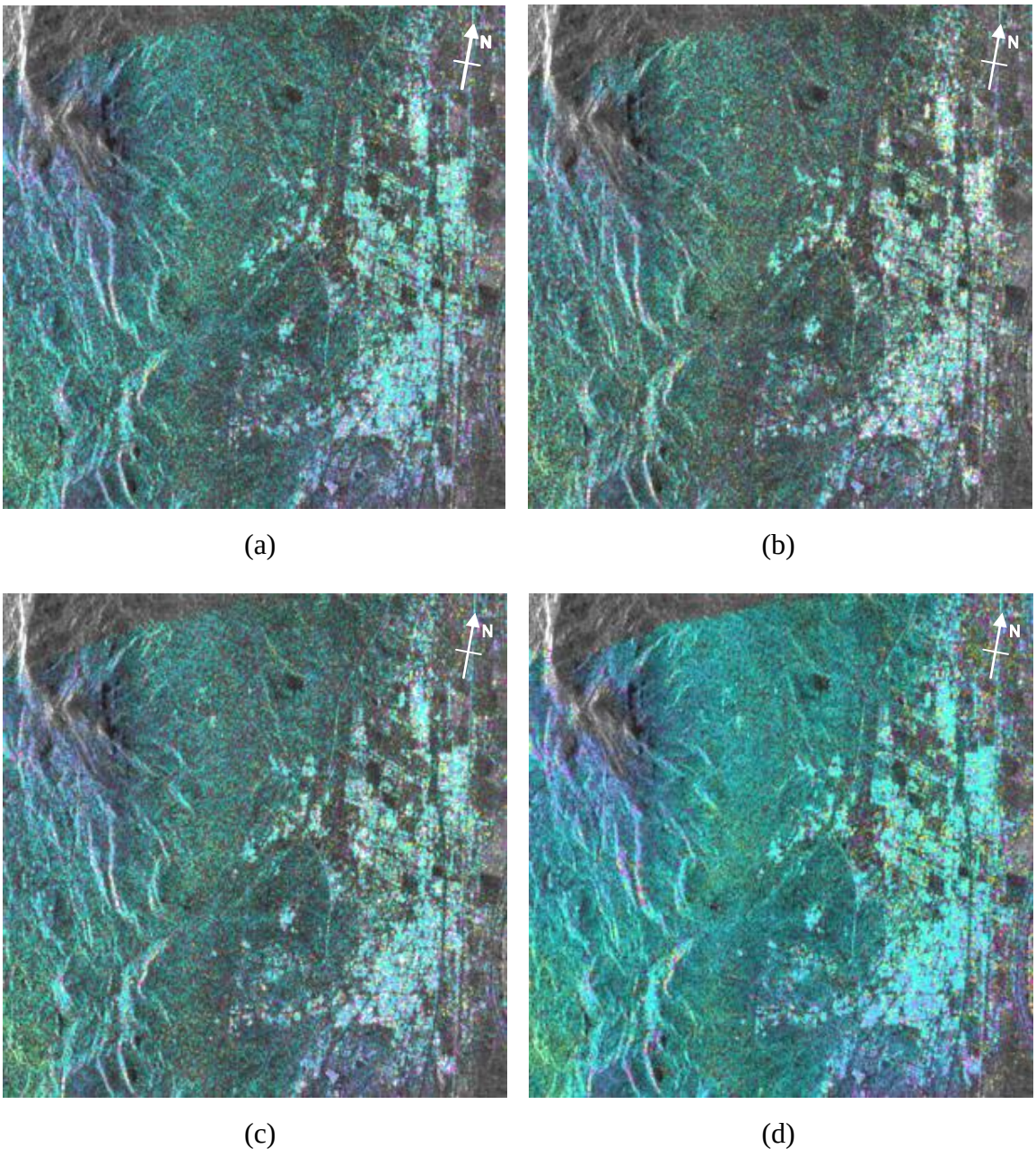


Figure 54. Final differential interferograms, 20080802_20080615. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

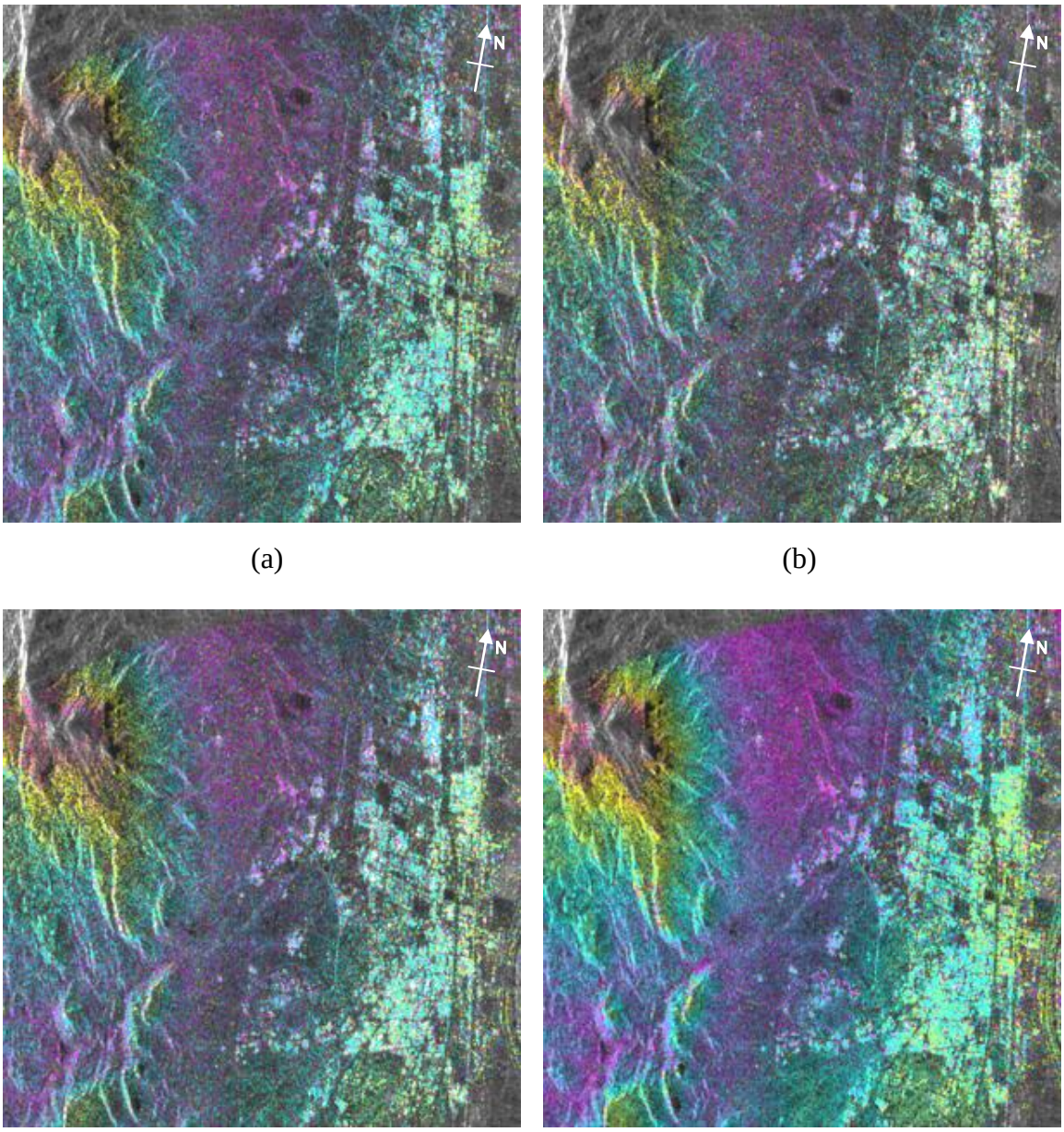


Figure 55. Final differential interferograms, 20080802_20080709. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

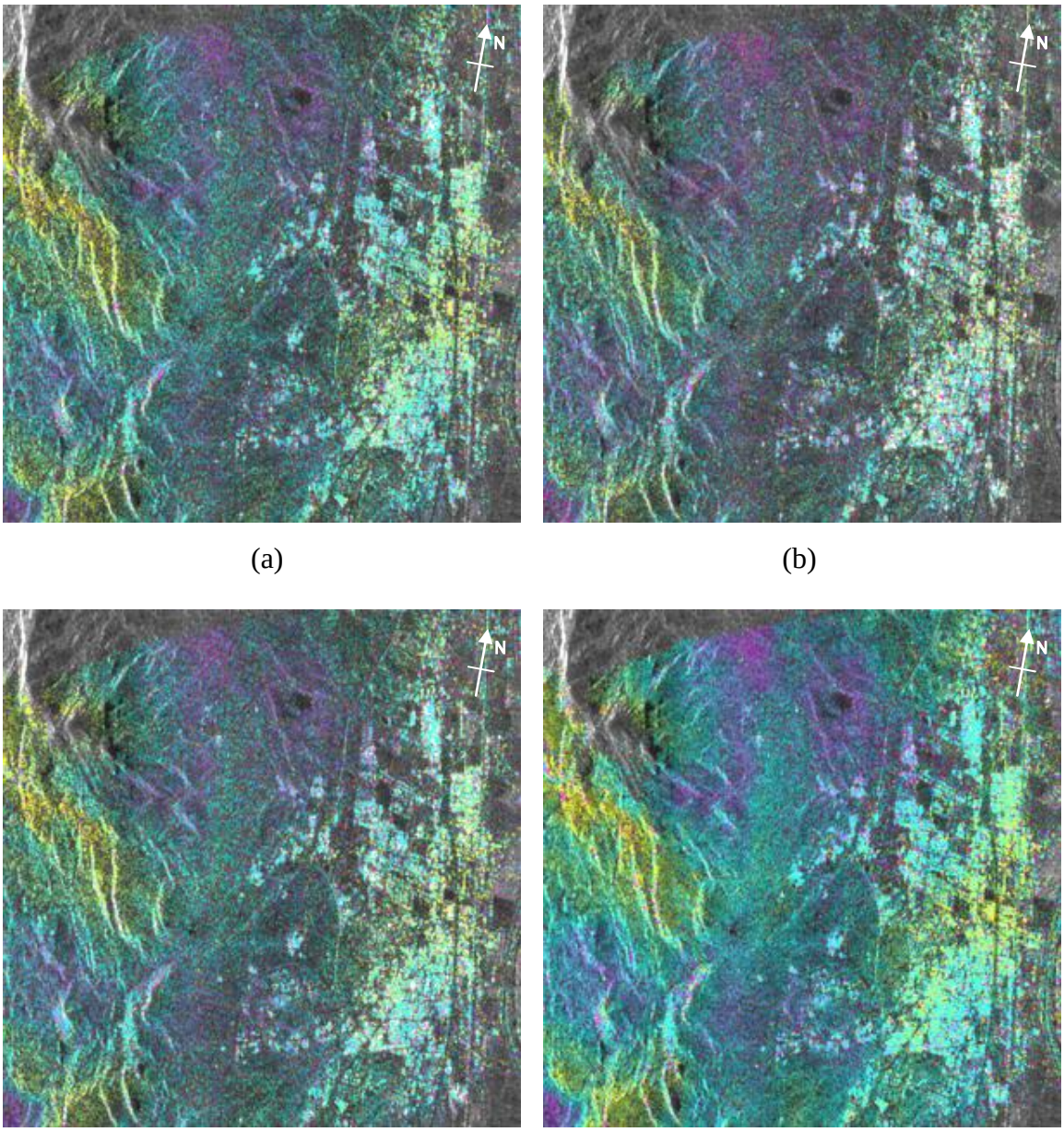


Figure 56. Final differential interferograms, 20080802_20080826. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

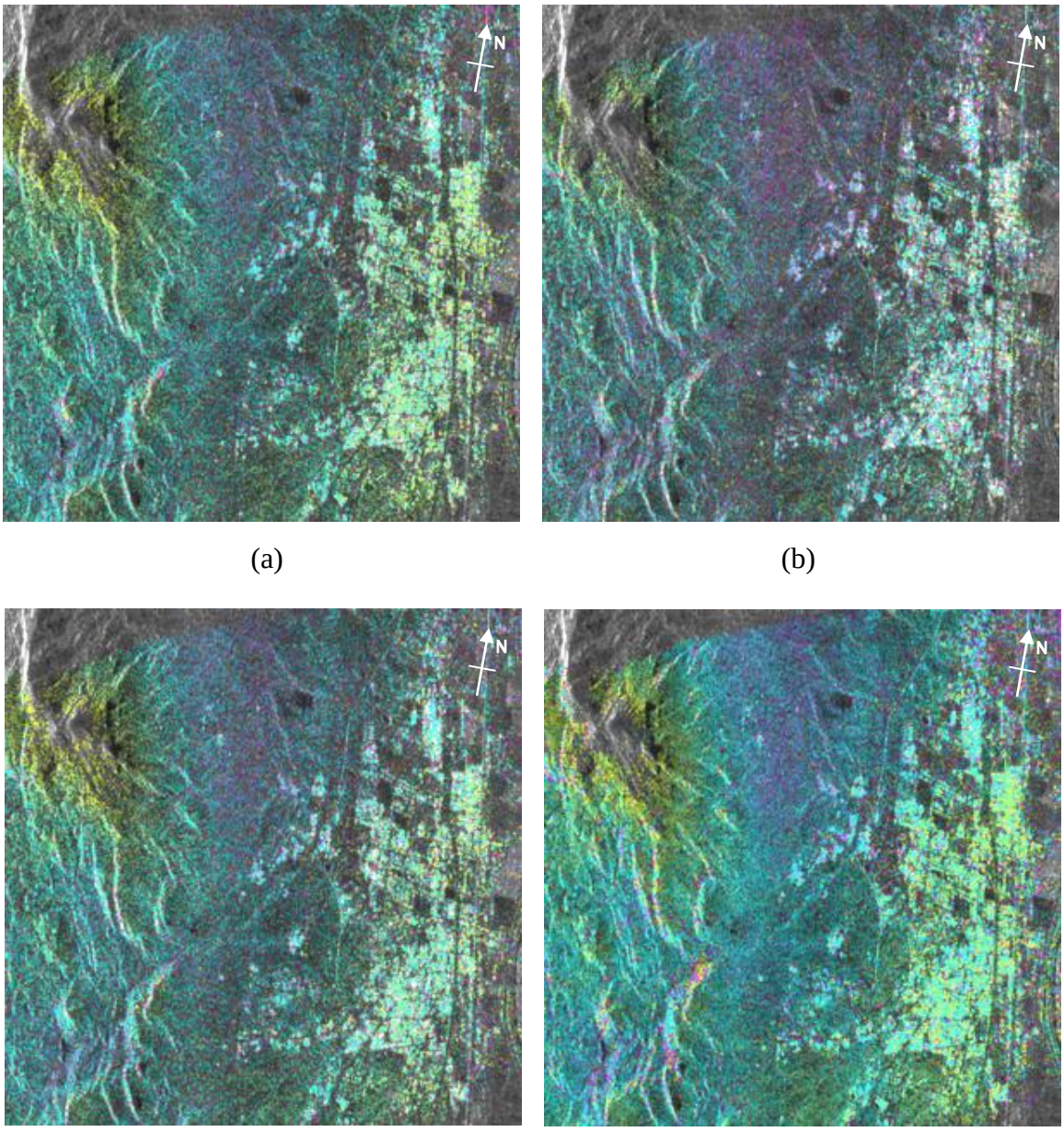


Figure 57. Final differential interferograms, 20080802_20080919. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

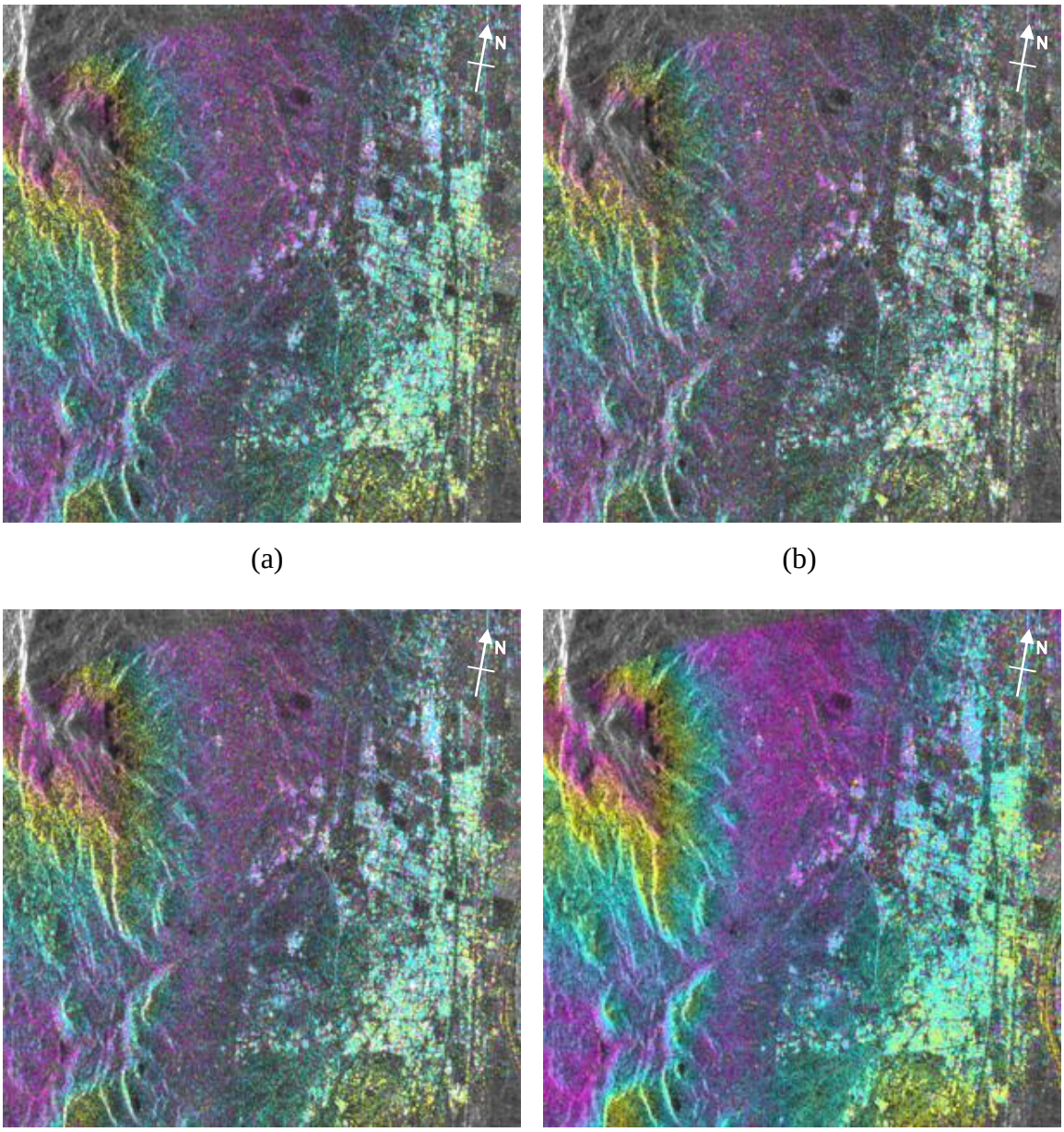


Figure 58. Final differential interferograms, 20080615_20080709. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

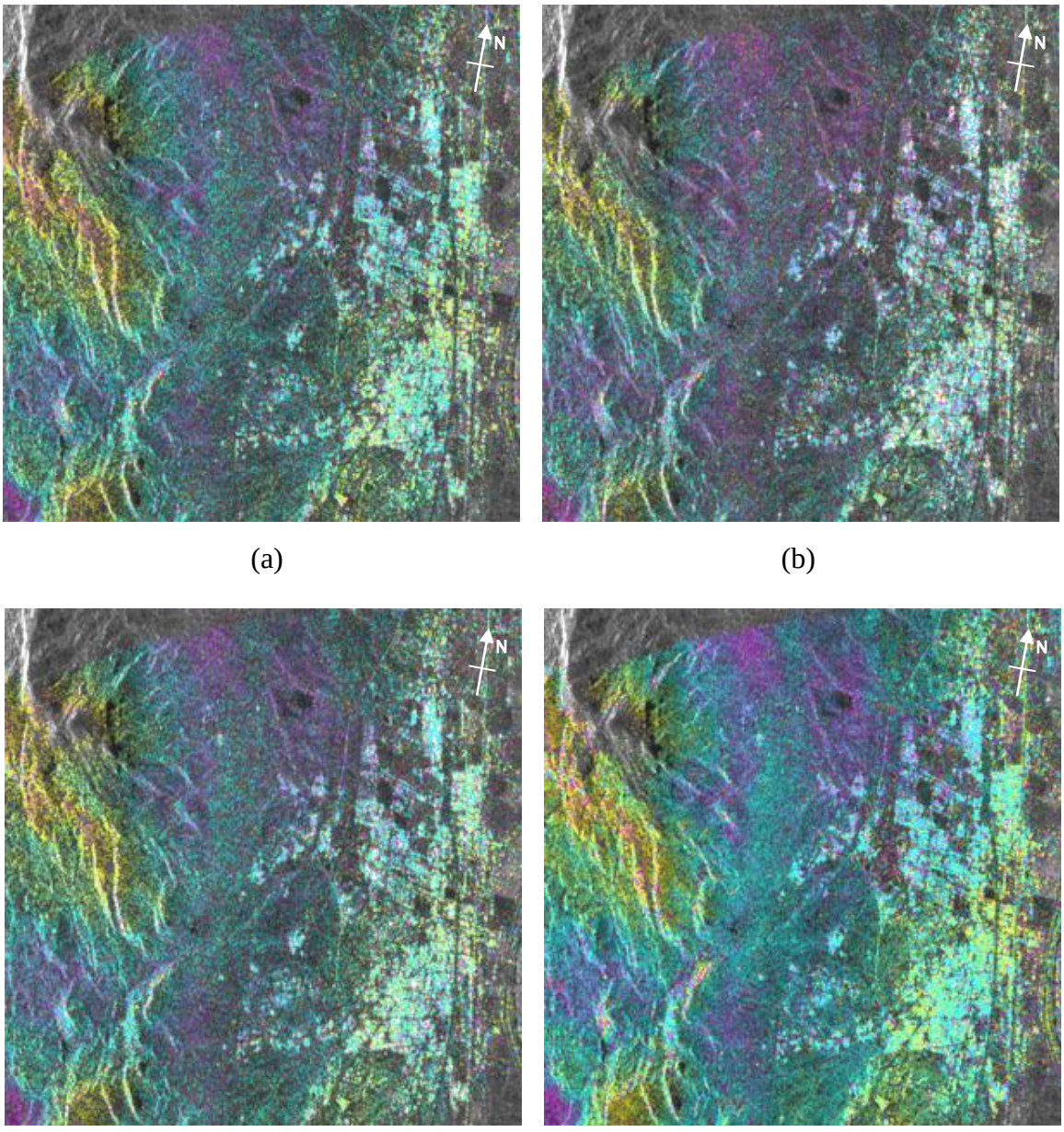


Figure 59. Final differential interferograms, 20080615_20080826. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

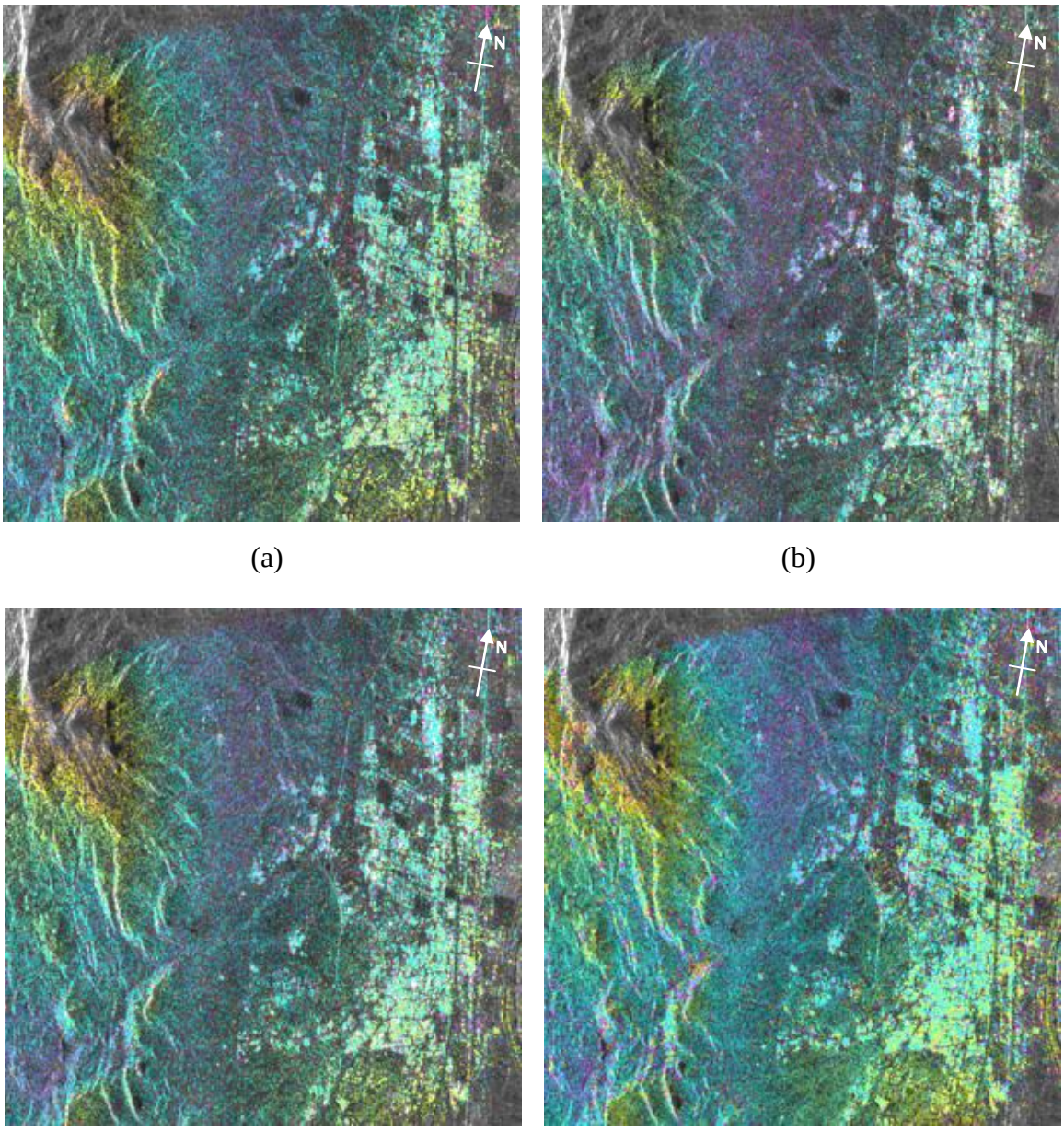


Figure 60. Final differential interferograms, 20080615_20080919. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

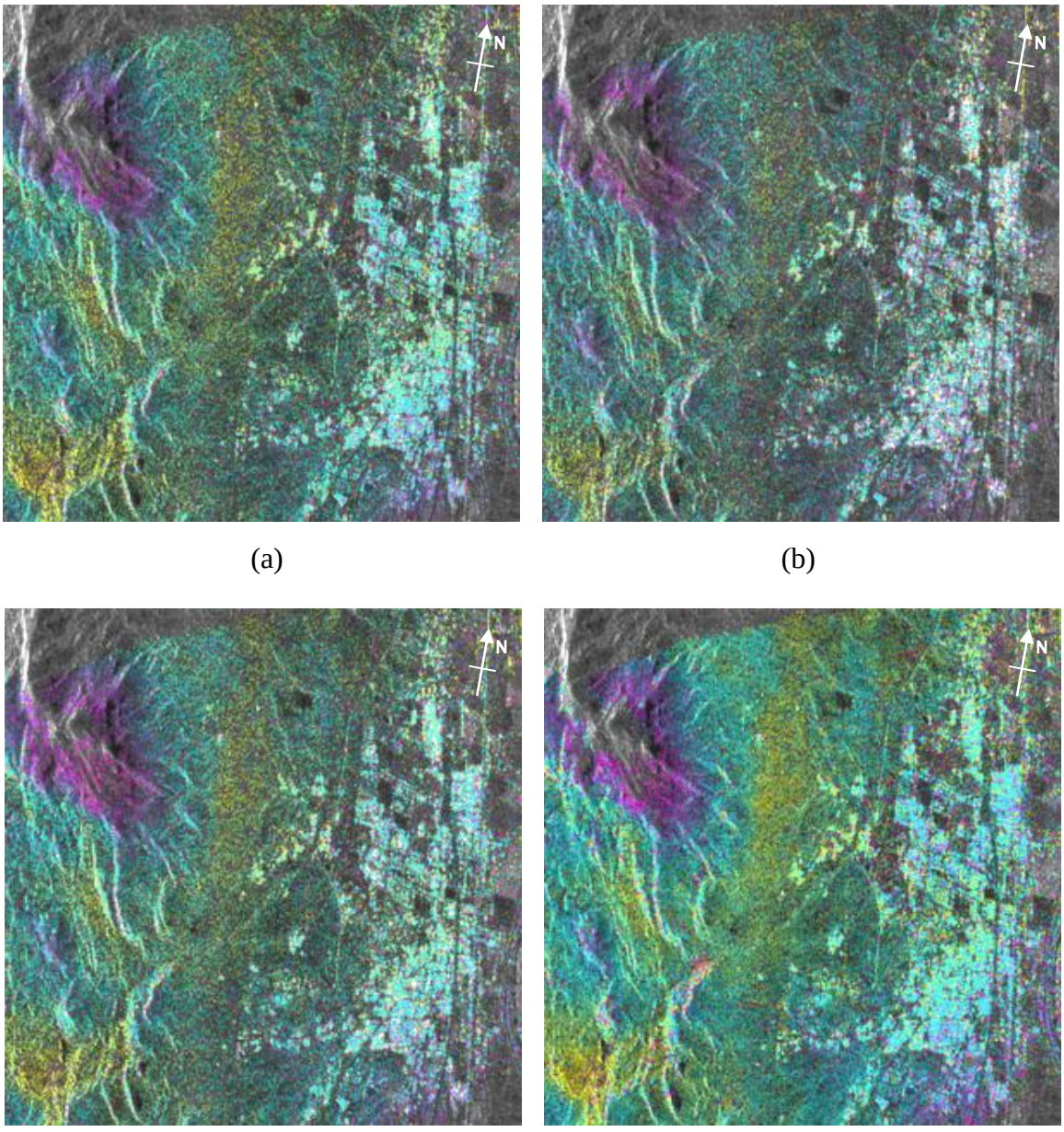


Figure 61. Final differential interferograms, 20080709_20080826. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

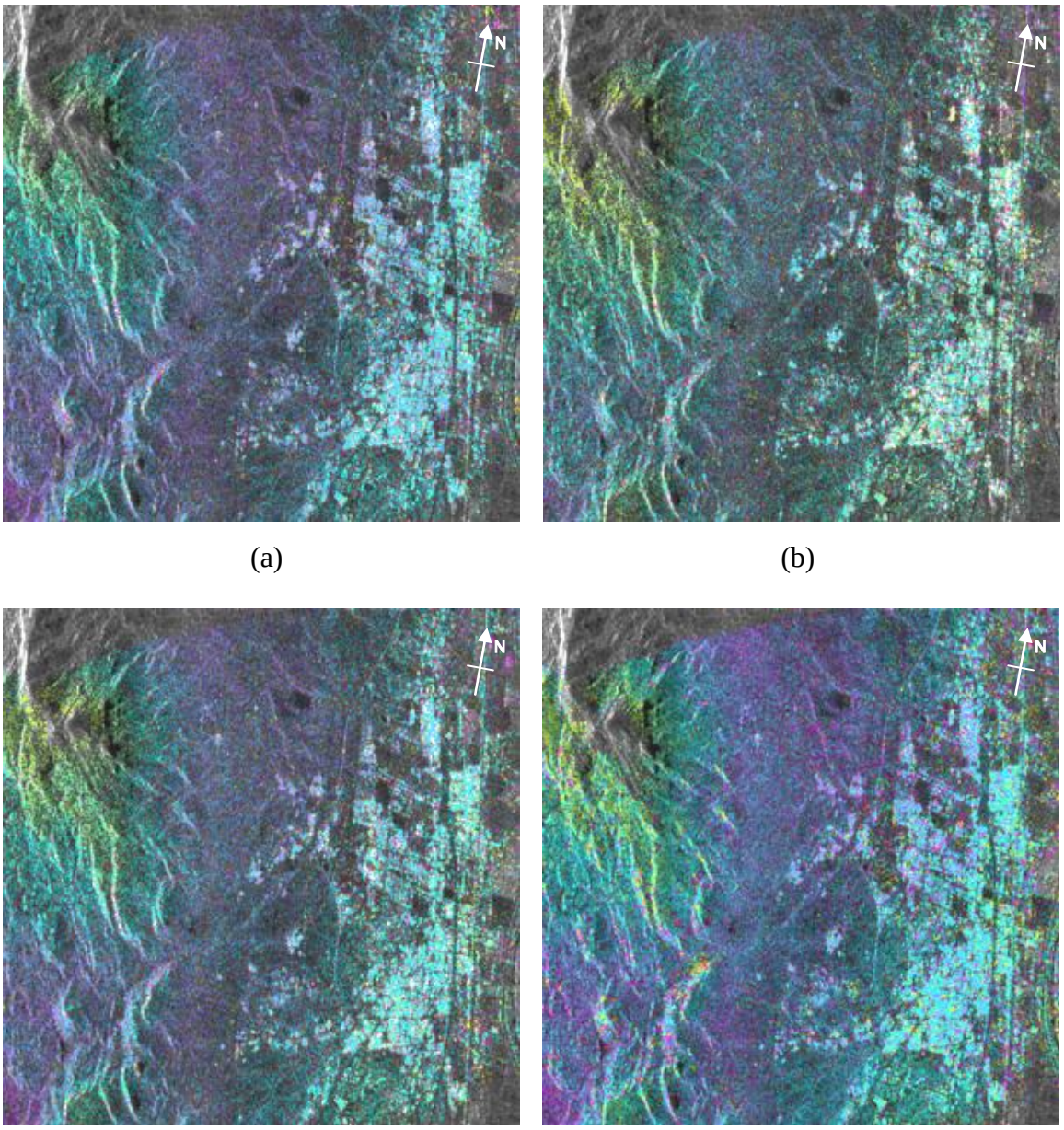


Figure 62. Final differential interferograms, 20080919_20080709. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

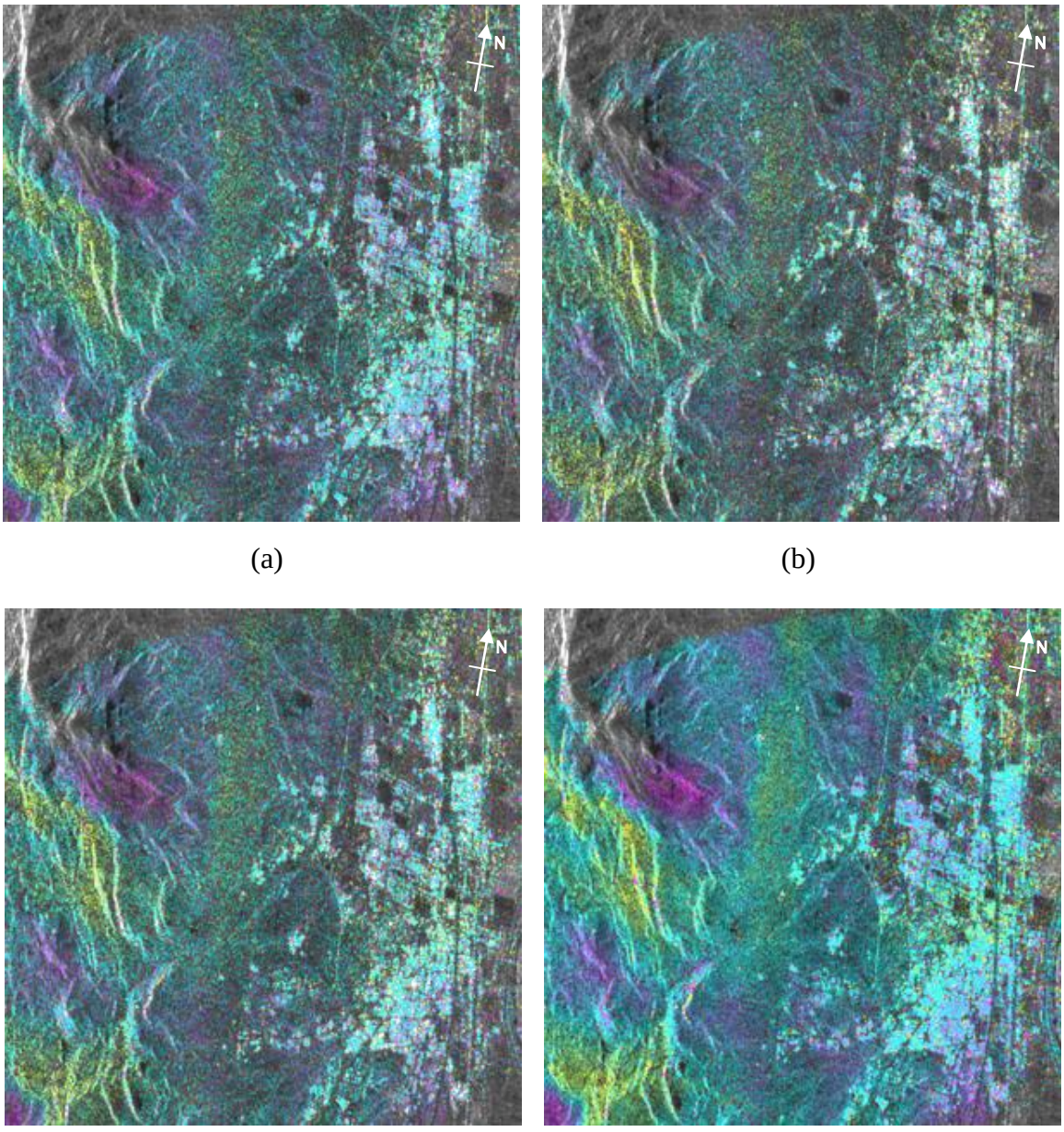


Figure 63. Final differential interferograms, 20080919_20080826. a) HH. b) HV. c) VV. d) PCO. One color cycle of phase corresponds to λ/2 (~28 mm).

7.5. Appendix E: Polarimetric Decompositions

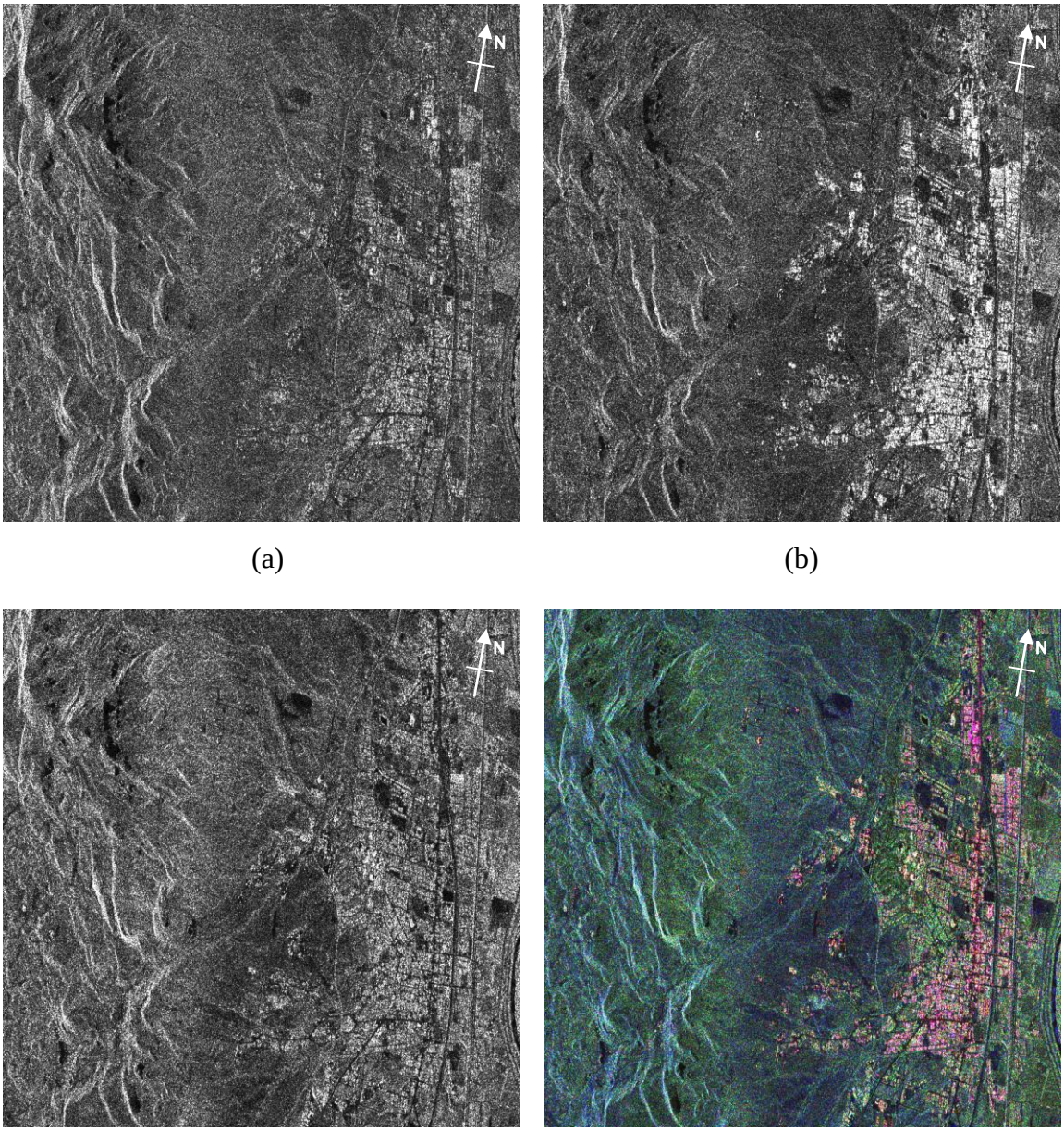


Figure 64. Pauli decomposition of 20080802 quad-pol collect . a) K1 SLC. b) K2 SLC. c) K3 SLC. d) RGB color composite: K1(red), K2(green), K3(blue).

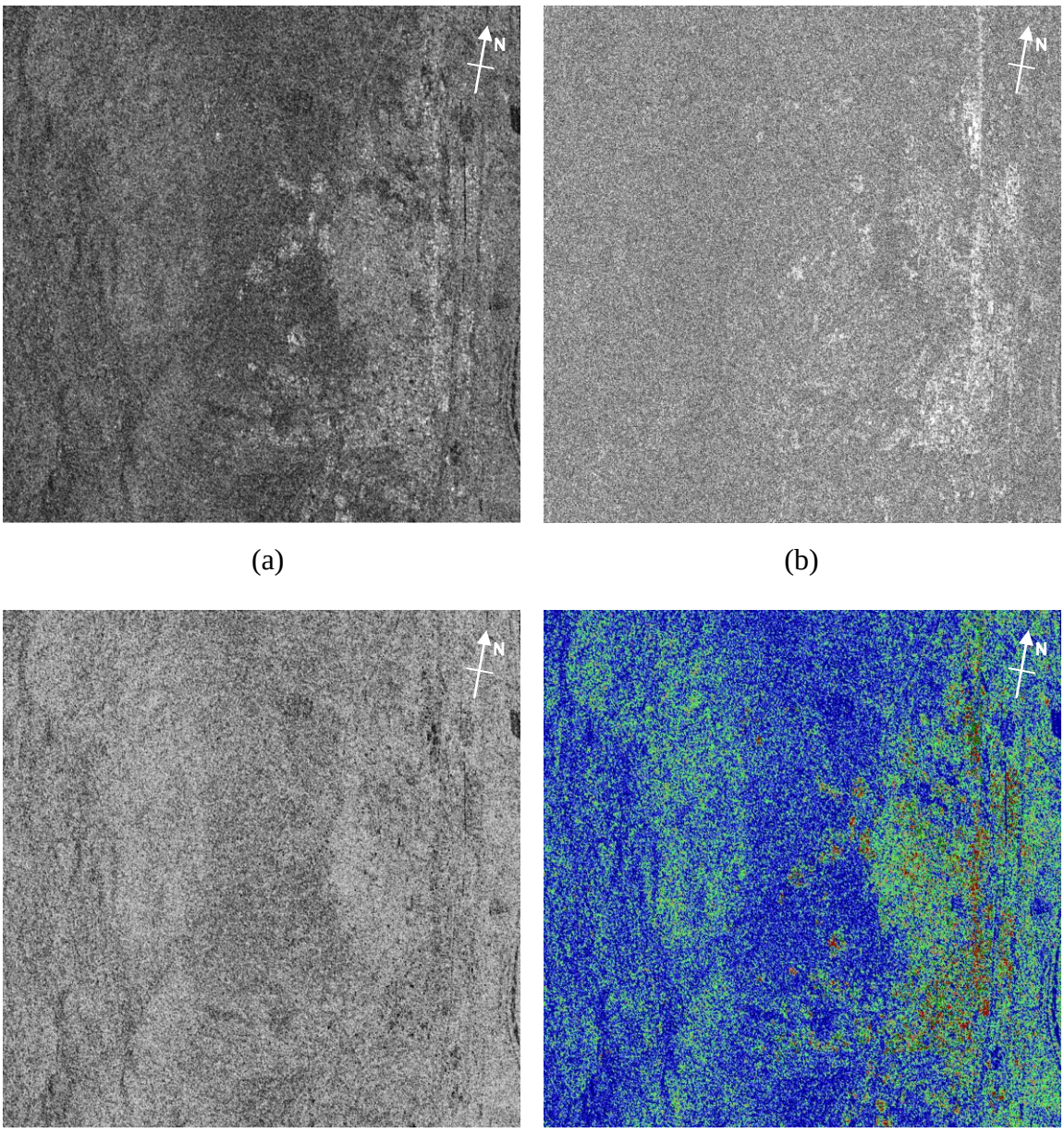


Figure 65. α-H decomposition of 20080802 quad-pol collect . a) α-angle. b) anisotropy. c) entropy. d) RGB classification map.

7.6. Appendix F: GAMMA Processing Scripts

main.sh

```
echo  ----- executing main.sh -----

mkdir rslc_chip
./rslc_chip.sh

mkdir plist
mkdir plist/cc
mkdir plist/msr
mkdir pwr
./spstat.sh
./plist.sh

mkdir pslc
./pslc.sh

mkdir pbase
./pbase.sh

mkdir pco_chip
mkdir pint0
./pco_chip.sh
./pint0.sh

mkdir phreg0
./phreg0.sh

mkdir pint1
./pint1.sh

mkdir base_ref
./base_ref.sh

mkdir pint2
./pint2.sh
```

rslc_chip.sh

```
echo ... subseting RSLCs

## COMMAND REFERENCE ##
# SLC_copy_all <SLC_tab> <SLC2_dir> <roff> <nr> <loff> <nl>

# Subsetting coregistered SLCs
SLC_copy_all tab/socorro_HH.slc_tab rslc_chip 1376 1200 4702 1200
SLC_copy_all tab/socorro_HV.slc_tab rslc_chip 1376 1200 4702 1200
SLC_copy_all tab/socorro_VV.slc_tab rslc_chip 1376 1200 4702 1200

# Subsetting DEM
copy geo/20080802HH.hgt rslc_chip/20080802HH.hgt 12992 4702 1200 0 0 5504 4800
```

pco_chip.sh

```
echo ... subsetting PCO interferograms and extracting point data

## COMMAND REFERENCE ##
# copy <infile> <outfile> <lbytes> [start] [nlines] [offset] [file_ldr] [offb] [nbyte]
# d2pt <f_in> <width> <plist> <rlks>  <azlks> <pdata> <rec_num> <type>

# Subsetting SARscape-generated PCO images using specified image coordinates
copy pco/20080802_20080615_max_dint pco_chip/20080802_20080615_max.dint 25984 4702 1200 0 0 11008 9600
copy pco/20080802_20080709_max_dint pco_chip/20080802_20080709_max.dint 25984 4702 1200 0 0 11008 9600
copy pco/20080802_20080826_max_dint pco_chip/20080802_20080826_max.dint 25984 4702 1200 0 0 11008 9600
copy pco/20080802_20080919_max_dint pco_chip/20080802_20080919_max.dint 25984 4702 1200 0 0 11008 9600
copy pco/20080615_20080709_max_dint pco_chip/20080615_20080709_max.dint 25984 4702 1200 0 0 11008 9600
copy pco/20080615_20080826_max_dint pco_chip/20080615_20080826_max.dint 25984 4702 1200 0 0 11008 9600
copy pco/20080615_20080919_max_dint pco_chip/20080615_20080919_max.dint 25984 4702 1200 0 0 11008 9600
copy pco/20080709_20080826_max_dint pco_chip/20080709_20080826_max.dint 25984 4702 1200 0 0 11008 9600
copy pco/20080919_20080709_max_dint pco_chip/20080919_20080709_max.dint 25984 4702 1200 0 0 11008 9600
copy pco/20080919_20080826_max_dint pco_chip/20080919_20080826_max.dint 25984 4702 1200 0 0 11008 9600

# Creating PCO point data stack by extracting the subset image values at the point list locations
d2pt pco_chip/20080802_20080615_max.dint 1200 plist/socorro_PCO.plist 1 1 pint0/socorro_PCO.pdint  1 0
d2pt pco_chip/20080802_20080709_max.dint 1200 plist/socorro_PCO.plist 1 1 pint0/socorro_PCO.pdint  2 0
d2pt pco_chip/20080802_20080826_max.dint 1200 plist/socorro_PCO.plist 1 1 pint0/socorro_PCO.pdint  3 0
d2pt pco_chip/20080802_20080919_max.dint 1200 plist/socorro_PCO.plist 1 1 pint0/socorro_PCO.pdint  4 0
d2pt pco_chip/20080615_20080709_max.dint 1200 plist/socorro_PCO.plist 1 1 pint0/socorro_PCO.pdint  5 0
d2pt pco_chip/20080615_20080826_max.dint 1200 plist/socorro_PCO.plist 1 1 pint0/socorro_PCO.pdint  6 0
d2pt pco_chip/20080615_20080919_max.dint 1200 plist/socorro_PCO.plist 1 1 pint0/socorro_PCO.pdint  7 0
d2pt pco_chip/20080709_20080826_max.dint 1200 plist/socorro_PCO.plist 1 1 pint0/socorro_PCO.pdint  8 0
d2pt pco_chip/20080919_20080709_max.dint 1200 plist/socorro_PCO.plist 1 1 pint0/socorro_PCO.pdint  9 0
d2pt pco_chip/20080919_20080826_max.dint 1200 plist/socorro_PCO.plist 1 1 pint0/socorro_PCO.pdint 10 0

# Subsetting SARscape-generated simulated unwrapped phase images using specified image coordinates
copy pco/20080802_20080615_sint pco_chip/20080802_20080615.sint 12992 4702 1200 0 0 5504 4800
copy pco/20080802_20080709_sint pco_chip/20080802_20080709.sint 12992 4702 1200 0 0 5504 4800
copy pco/20080802_20080826_sint pco_chip/20080802_20080826.sint 12992 4702 1200 0 0 5504 4800
copy pco/20080802_20080919_sint pco_chip/20080802_20080919.sint 12992 4702 1200 0 0 5504 4800
copy pco/20080615_20080709_sint pco_chip/20080615_20080709.sint 12992 4702 1200 0 0 5504 4800
copy pco/20080615_20080826_sint pco_chip/20080615_20080826.sint 12992 4702 1200 0 0 5504 4800
copy pco/20080615_20080919_sint pco_chip/20080615_20080919.sint 12992 4702 1200 0 0 5504 4800
copy pco/20080709_20080826_sint pco_chip/20080709_20080826.sint 12992 4702 1200 0 0 5504 4800
copy pco/20080919_20080709_sint pco_chip/20080919_20080709.sint 12992 4702 1200 0 0 5504 4800
copy pco/20080919_20080826_sint pco_chip/20080919_20080826.sint 12992 4702 1200 0 0 5504 4800

# Creating simulated phase point data stack by extracting the subset image values at the point list locations
d2pt pco_chip/20080802_20080615.sint 1200 plist/socorro_PCO.plist 1 1 pco_chip/socorro_PCO.psint_orig  1 2
d2pt pco_chip/20080802_20080709.sint 1200 plist/socorro_PCO.plist 1 1 pco_chip/socorro_PCO.psint_orig  2 2
d2pt pco_chip/20080802_20080826.sint 1200 plist/socorro_PCO.plist 1 1 pco_chip/socorro_PCO.psint_orig  3 2
d2pt pco_chip/20080802_20080919.sint 1200 plist/socorro_PCO.plist 1 1 pco_chip/socorro_PCO.psint_orig  4 2
d2pt pco_chip/20080615_20080709.sint 1200 plist/socorro_PCO.plist 1 1 pco_chip/socorro_PCO.psint_orig  5 2
d2pt pco_chip/20080615_20080826.sint 1200 plist/socorro_PCO.plist 1 1 pco_chip/socorro_PCO.psint_orig  6 2
d2pt pco_chip/20080615_20080919.sint 1200 plist/socorro_PCO.plist 1 1 pco_chip/socorro_PCO.psint_orig  7 2
d2pt pco_chip/20080709_20080826.sint 1200 plist/socorro_PCO.plist 1 1 pco_chip/socorro_PCO.psint_orig  8 2
d2pt pco_chip/20080919_20080709.sint 1200 plist/socorro_PCO.plist 1 1 pco_chip/socorro_PCO.psint_orig  9 2
d2pt pco_chip/20080919_20080826.sint 1200 plist/socorro_PCO.plist 1 1 pco_chip/socorro_PCO.psint_orig 10 2

# Negate extracted values so that they are consistant with GAMMA simulated phase processing
```

lin_comb_pt plist/socorro_PCO.plist - pco_chip/socorro_PCO.psint_orig - pco_chip/socorro_PCO.psint_orig - pint0/socorro_PCO.psint - 0. 0. -1. 2

spstat.sh

echo ... generating point candidate lists for each polarization

Command Reference
sp_stat <SLC> <pwr> <cc> <MSR> <plist> <width> [PWR_min] [CC_min] [MSR_min] [rlks] [azlks] [roff] [loff] [nr] [nl] [bx] [by] [type] [r_ovr]

Generates a point list based on low spectral diversity
sp_stat rslc_chip/20080802HH.rslc pwr/20080802HH.pwr plist/cc/20080802HH.cc plist/msr/20080802HH.msr plist/20080802HH.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1
sp_stat rslc_chip/20080615HH.rslc - plist/cc/20080615HH.cc plist/msr/20080615HH.msr plist/20080615HH.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1
sp_stat rslc_chip/20080709HH.rslc - plist/cc/20080709HH.cc plist/msr/20080709HH.msr plist/20080709HH.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1
sp_stat rslc_chip/20080826HH.rslc - plist/cc/20080826HH.cc plist/msr/20080826HH.msr plist/20080826HH.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1
sp_stat rslc_chip/20080919HH.rslc - plist/cc/20080919HH.cc plist/msr/20080919HH.msr plist/20080919HH.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1

sp_stat rslc_chip/20080802HV.rslc pwr/20080802HV.pwr plist/cc/20080802HV.cc plist/msr/20080802HV.msr plist/20080802HV.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1
sp_stat rslc_chip/20080615HV.rslc - plist/cc/20080615HV.cc plist/msr/20080615HV.msr plist/20080615HV.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1
sp_stat rslc_chip/20080709HV.rslc - plist/cc/20080709HV.cc plist/msr/20080709HV.msr plist/20080709HV.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1
sp_stat rslc_chip/20080826HV.rslc - plist/cc/20080826HV.cc plist/msr/20080826HV.msr plist/20080826HV.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1
sp_stat rslc_chip/20080919HV.rslc - plist/cc/20080919HV.cc plist/msr/20080919HV.msr plist/20080919HV.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1

sp_stat rslc_chip/20080802VV.rslc pwr/20080802VV.pwr plist/cc/20080802VV.cc plist/msr/20080802VV.msr plist/20080802VV.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1
sp_stat rslc_chip/20080615VV.rslc - plist/cc/20080615VV.cc plist/msr/20080615VV.msr plist/20080615VV.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1
sp_stat rslc_chip/20080709VV.rslc - plist/cc/20080709VV.cc plist/msr/20080709VV.msr plist/20080709VV.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1
sp_stat rslc_chip/20080826VV.rslc - plist/cc/20080826VV.cc plist/msr/20080826VV.msr plist/20080826VV.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1
sp_stat rslc_chip/20080919VV.rslc - plist/cc/20080919VV.cc plist/msr/20080919VV.msr plist/20080919VV.plist 1200 0.0 0.4 1.0 4 4 0 0 - - - - 1 1

plist.sh

```
echo ... generating initial point candidate list


# Merges plists retaining points that occur more than once
merge_pt tab/socorro_HH.plist_tab plist/socorro_HH.plist 2 0 0
merge_pt tab/socorro_HV.plist_tab plist/socorro_HV.plist 2 0 0
merge_pt tab/socorro_VV.plist_tab plist/socorro_VV.plist 2 0 0


# Create plist for quad-pol interferometry case
merge_pt tab/socorro_PCO.plist_tab plist/socorro_PCO.plist 1 0 0


# Creates rasters
raspwr pwr/20080802HH.pwr 1200 1 - 1 1 0.25 .35 - pwr/20080802HH.pwr.ras
raspwr pwr/20080802HV.pwr 1200 1 - 1 1 0.25 .35 - pwr/20080802HV.pwr.ras
raspwr pwr/20080802VV.pwr 1200 1 - 1 1 0.25 .35 - pwr/20080802VV.pwr.ras

ras_pt plist/socorro_HH.plist - pwr/20080802HH.pwr.ras plist/socorro_HH.ras 1 1 255 0 0 3 0 0
ras_pt plist/socorro_HV.plist - pwr/20080802HV.pwr.ras plist/socorro_HV.ras 1 1 255 0 0 3 0 0
ras_pt plist/socorro_VV.plist - pwr/20080802VV.pwr.ras plist/socorro_VV.ras 1 1 255 0 0 3 0 0
ras_pt plist/socorro_PCO.plist - pwr/20080802HH.pwr.ras plist/socorro_PCO.ras 1 1 255 0 0 3 0 0
```

pslc.sh

```
echo ... extracting SLC point data

#$ SLC2pt <SLC_tab> <plist> <pmask> <pSLC_par>  <pSLC> <SLC_rec_num>

# extract point data from SLCs using plist for a given polarization
SLC2pt tab/socorro_HH.chip_tab plist/socorro_HH.plist - pslc/socorro_HH.pslc_par pslc/socorro_HH.pslc -
SLC2pt tab/socorro_HV.chip_tab plist/socorro_HV.plist - pslc/socorro_HV.pslc_par pslc/socorro_HV.pslc -
SLC2pt tab/socorro_VV.chip_tab plist/socorro_VV.plist - pslc/socorro_VV.pslc_par pslc/socorro_VV.pslc -
SLC2pt tab/socorro_HH.chip_tab plist/socorro_PCO.plist - pslc/socorro_PCO.pslc_par pslc/socorro_PCO.pslc -
```

pbase.sh

```
echo ... estimating perpendicular baselines from orbital parameters

## COMMAND REFERENCE ##
# base_orbit_pt <pSLC_par> <itab> <rec_num>  <pbase>
#  base_calc  <SLC_tab>  <rslc_par>  <gr_file>    <bperp_file>  [itab]  [itab_mode]  [bperp_max]  [delta_T_min]
[delta_T_max]

# Calculating initial baseline estimates from orbital state vectors
base_orbit_pt pslc/socorro_HH.pslc_par tab/socorro_HH.itab - pbase/socorro.pbase

cp pbase/socorro.pbase pbase/socorro_HH.pbase
```

```
cp pbase/socorro.pbase pbase/socorro_HV.pbase
cp pbase/socorro.pbase pbase/socorro_VV.pbase
cp pbase/socorro.pbase pbase/socorro_PCO.pbase
```

pint0.sh

```
echo ... generating initial differential interferometric point data stacks

## COMMAND REFERENCE ##
# data2pt <f_in> <par_in> <plist> <SLC_par>  <pdata> <rec_num> <type>
# intf_pt <plist> <pmask> <itab> <rec_num>  <pSLC> <pint> <type> [pSLC_par]
# pdisdt_pwr24 <plist> <pmask> <SLC_par>  <pdata> <rec_num> <par_out> <mli> <cycle>  [radius]
# ccs_pt <plist> <pmask> <SLC_par> <pdata_in>  <pccs> [rec_num] [type] [r_max] [w_func] [np_max]
# mcf_pt <plist> <pmask> <pint> <rec_num>  <pwgt> <mask> <punw> [rps] [azps] [np_ref] [phref_flag] <rlks>
<azlks>
# pdisdt_pwr24 <plist> <pmask> <SLC_par>  <pdata> <rec_num> <par_out> <mli> <cycle>  [radius]
# prasdt_pwr24 <plist> <pmask> <SLC_par>  <pdata> <rec_num> <par_out> <mli> <cycle>  [radius]

# Converts DEM to vector format
data2pt rslc_chip/20080802HH.hgt rslc_chip/20080802HH.rslc.par plist/socorro_HH.plist
rslc_chip/20080802HH.rslc.par pint0/socorro_HH.pdem 1 2
data2pt rslc_chip/20080802HH.hgt rslc_chip/20080802HH.rslc.par plist/socorro_HV.plist
rslc_chip/20080802HH.rslc.par pint0/socorro_HV.pdem 1 2
data2pt rslc_chip/20080802HH.hgt rslc_chip/20080802HH.rslc.par plist/socorro_VV.plist
rslc_chip/20080802HH.rslc.par pint0/socorro_VV.pdem 1 2
data2pt rslc_chip/20080802HH.hgt rslc_chip/20080802HH.rslc.par plist/socorro_PCO.plist
rslc_chip/20080802HH.rslc.par pint0/socorro_PCO.pdem 1 2

# Generates point interferogram data for each polarization
intf_pt plist/socorro_HH.plist - tab/socorro_HH.itab - pslc/socorro_HH.pslc pint0/socorro_HH.pint 1
intf_pt plist/socorro_HV.plist - tab/socorro_HV.itab - pslc/socorro_HV.pslc pint0/socorro_HV.pint 1
intf_pt plist/socorro_VV.plist - tab/socorro_VV.itab - pslc/socorro_VV.pslc pint0/socorro_VV.pint 1

# Calculates initial phase model based on perpendicular baselines and DEM
phase_sim_pt plist/socorro_HH.plist - pslc/socorro_HH.pslc_par - tab/socorro_HH.itab - pbase/socorro_HH.pbase
pint0/socorro_HH.pdem pint0/socorro_HH.psim_unw0 - 0 0
phase_sim_pt plist/socorro_HV.plist - pslc/socorro_HH.pslc_par - tab/socorro_HV.itab - pbase/socorro_HV.pbase
pint0/socorro_HV.pdem pint0/socorro_HV.psim_unw0 - 0 0
phase_sim_pt plist/socorro_VV.plist - pslc/socorro_HH.pslc_par - tab/socorro_VV.itab - pbase/socorro_VV.pbase
pint0/socorro_VV.pdem pint0/socorro_VV.psim_unw0 - 0 0
phase_sim_pt plist/socorro_PCO.plist - pslc/socorro_PCO.pslc_par - tab/socorro_PCO.itab - pbase/socorro_PCO.pbase
pint0/socorro_PCO.pdem pint0/socorro_PCO.psim_unw0 - 0 0

# For PCO, add back in the simulated phase used during SARscape PCO processing so that Gamma simulated phase
can be used
sub_phase_pt plist/socorro_PCO.plist - pint0/socorro_PCO.pdint - pint0/socorro_PCO.psint pint0/socorro_PCO.pint 1
1

# Generating raster images of the complex interferograms
prasmph_pwr24 plist/socorro_HH.plist - rslc_chip/20080802HH.rslc.par pint0/socorro_HH.pint -
rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 3
prasmph_pwr24 plist/socorro_HV.plist - rslc_chip/20080802HH.rslc.par pint0/socorro_HV.pint -
rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 3
prasmph_pwr24 plist/socorro_VV.plist - rslc_chip/20080802HH.rslc.par pint0/socorro_VV.pint -
rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 3
```

```
prasmph_pwr24 plist/socorro_PCO.plist - rslc_chip/20080802HH.rslc.par pint0/socorro_PCO.pint -
rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 3

# Subtracting initial phase model from the complex interferograms
sub_phase_pt plist/socorro_HH.plist - pint0/socorro_HH.pint - pint0/socorro_HH.psim_unw0 pint0/socorro_HH.pdiff0
1
sub_phase_pt plist/socorro_HV.plist - pint0/socorro_HV.pint - pint0/socorro_HV.psim_unw0 pint0/socorro_HV.pdiff0
1
sub_phase_pt plist/socorro_VV.plist - pint0/socorro_VV.pint - pint0/socorro_VV.psim_unw0 pint0/socorro_VV.pdiff0
1
sub_phase_pt plist/socorro_PCO.plist - pint0/socorro_PCO.pint - pint0/socorro_PCO.psim_unw0
pint0/socorro_PCO.pdiff0 1

# Generating raster images of the differential interferograms
prasmph_pwr24 plist/socorro_HH.plist - rslc_chip/20080802HH.rslc.par pint0/socorro_HH.pdiff0 -
rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 3
prasmph_pwr24 plist/socorro_HV.plist - rslc_chip/20080802HH.rslc.par pint0/socorro_HV.pdiff0 -
rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 3
prasmph_pwr24 plist/socorro_VV.plist - rslc_chip/20080802HH.rslc.par pint0/socorro_VV.pdiff0 -
rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 3
prasmph_pwr24 plist/socorro_PCO.plist - rslc_chip/20080802HH.rslc.par pint0/socorro_PCO.pdiff0 -
rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 3

# Manually select a reference point for automated regression analysis
dis_ipta plist/socorro_HH.plist - pslc/socorro_HH.pslc_par - tab/socorro_HH.itab pbase/socorro.pbase 0
pint0/socorro_HH.pdiff0 1 pint0/socorro_HH.pdiff0.03.ras 30 - 0.01 1
dis_ipta plist/socorro_HV.plist - pslc/socorro_HH.pslc_par - tab/socorro_HV.itab pbase/socorro.pbase 0
pint0/socorro_HV.pdiff0 1 pint0/socorro_HV.pdiff0.03.ras 30 - 0.01 1
dis_ipta plist/socorro_VV.plist - pslc/socorro_HH.pslc_par - tab/socorro_VV.itab pbase/socorro.pbase 0
pint0/socorro_VV.pdiff0 1 pint0/socorro_VV.pdiff0.03.ras 30 - 0.01 1
dis_ipta plist/socorro_PCO.plist - pslc/socorro_PCO.pslc_par - tab/socorro_PCO.itab pbase/socorro_PCO.pbase 0
pint0/socorro_PCO.pdiff0 1 pint0/socorro_PCO.pdiff0.03.ras 30 - 0.01 1

# Pause until editing is done
echo STOP: MANUALLY SET REFERENCE POINT NUMBER IN SCRIPTS BEFORE CONTINUING
gedit jg_phreg0.sh
gedit jg_base_ref.sh
read -p "press any key to continue... " -n 1 -s

# Performing MCF point-wise phase unwrapping
mcf_pt plist/socorro_HH.plist - pint0/socorro_HH.pdiff0 - - - pint0/socorro_HH.punw0
mcf_pt plist/socorro_HV.plist - pint0/socorro_HV.pdiff0 - - - pint0/socorro_HV.punw0
mcf_pt plist/socorro_VV.plist - pint0/socorro_VV.pdiff0 - - - pint0/socorro_VV.punw0
mcf_pt plist/socorro_PCO.plist - pint0/socorro_PCO.pdiff0 - - - pint0/socorro_PCO.punw0

# Generating raster images of the unwrapped phase
prasdt_pwr24 plist/socorro_HH.plist - rslc_chip/20080802HH.rslc.par pint0/socorro_HH.punw0 -
rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_HV.plist - rslc_chip/20080802HH.rslc.par pint0/socorro_HV.punw0 -
rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_VV.plist - rslc_chip/20080802HH.rslc.par pint0/socorro_VV.punw0 -
rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_PCO.plist - rslc_chip/20080802HH.rslc.par pint0/socorro_PCO.punw0 -
rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
```

phreg0.sh

echo ... performing initial phase regression analysis

COMMAND REFERENCE
mcf_pt <plist> <pmask> <pint> <rec_num> <pwgt> <mask> <punw> [rps] [azps] [np_ref] [phref_flag] <rlks> <azlks>
dis_ipta <plist> <pmask> <pSLC_par> <ppos> <itab> <pbase> <bflag> <pdiff> <pdiff_type> <ras> [dh_max] [def_min] [def_max] [model] [bmax] [dtmax] [mag] [win_sz]
multi_def_pt <plist> <pmask> <pSLC_par> <ppos> <itab> <pbase> <bflag> <pdiff> <pdiff_type> <np_ref> <pres> <pdh> <pdef> <punw> <psigma> <pmask_out> [dh_max] [def_min] [def_max] [rpatch] [sigma_max] [sigma_max2] [model] [bmax] [dtmax]

Perform automated phase regression analysis using global ref point and local patch ref points
def_mod_pt plist/socorro_HH.plist - pslc/socorro_HH.pslc_par - tab/socorro_HH.itab pbase/socorro.pbase 0 pint0/socorro_HH.punw0 0 12901 phreg0/socorro_HH.pres1 phreg0/socorro_HH.pdh1 phreg0/socorro_HH.pddef1 phreg0/socorro_HH.punw1 phreg0/socorro_HH.psigma1 phreg0/socorro_HH.pmask1 30 - 0.01 20.0 1 - -
def_mod_pt plist/socorro_HV.plist - pslc/socorro_HH.pslc_par - tab/socorro_HV.itab pbase/socorro.pbase 0 pint0/socorro_HV.punw0 0 10554 phreg0/socorro_HV.pres1 phreg0/socorro_HV.pdh1 phreg0/socorro_HV.pddef1 phreg0/socorro_HV.punw1 phreg0/socorro_HV.psigma1 phreg0/socorro_HV.pmask1 30 - 0.01 20.0 1 - -
def_mod_pt plist/socorro_VV.plist - pslc/socorro_HH.pslc_par - tab/socorro_VV.itab pbase/socorro.pbase 0 pint0/socorro_VV.punw0 0 12783 phreg0/socorro_VV.pres1 phreg0/socorro_VV.pdh1 phreg0/socorro_VV.pddef1 phreg0/socorro_VV.punw1 phreg0/socorro_VV.psigma1 phreg0/socorro_VV.pmask1 30 - 0.01 20.0 1 - -
def_mod_pt plist/socorro_PCO.plist - pslc/socorro_PCO.pslc_par - tab/socorro_PCO.itab pbase/socorro_PCO.pbase 0 pint0/socorro_PCO.punw0 0 33220 phreg0/socorro_PCO.pres1 phreg0/socorro_PCO.pdh1 phreg0/socorro_PCO.pddef1 phreg0/socorro_PCO.punw1 phreg0/socorro_PCO.psigma1 phreg0/socorro_PCO.pmask1 30 - 0.01 20.0 1 - -

Display residual phase results for manually detection of unwrapping errors
pdisdt_pwr24 plist/socorro_HH.plist phreg0/socorro_HH.pmask1 rslc_chip/20080802HH.rslc.par phreg0/socorro_HH.pres1 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 12.6 3
pdisdt_pwr24 plist/socorro_HV.plist phreg0/socorro_HV.pmask1 rslc_chip/20080802HH.rslc.par phreg0/socorro_HV.pres1 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 12.6 3
pdisdt_pwr24 plist/socorro_VV.plist phreg0/socorro_VV.pmask1 rslc_chip/20080802HH.rslc.par phreg0/socorro_VV.pres1 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 12.6 3
pdisdt_pwr24 plist/socorro_PCO.plist phreg0/socorro_PCO.pmask1 rslc_chip/20080802HH.rslc.par phreg0/socorro_PCO.pres1 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 12.6 3

Manually edit itab to exclude pairs with unwrapping errors
cp tab/socorro_HH.itab tab/socorro_HH.itab_rev1
cp tab/socorro_HV.itab tab/socorro_HV.itab_rev1
cp tab/socorro_VV.itab tab/socorro_VV.itab_rev1
cp tab/socorro_PCO.itab tab/socorro_PCO.itab_rev1

```
gedit tab/socorro_HH.itab_rev1
gedit tab/socorro_HV.itab_rev1
gedit tab/socorro_VV.itab_rev1
gedit tab/socorro_PCO.itab_rev1

echo STOP: MANUALLY REVIEW UNWRAPPED PHASE AND EDIT VALIDITY FLAGS IN ITAB BEFORE CONTINUING
read -p "press any key to continue... " -n 1 -s

# Re-running regression analysis for correctly unwrapped layers and w/ point mask
def_mod_pt plist/socorro_HH.plist phreg0/socorro_HH.pmask1 pslc/socorro_HH.pslc_par - tab/socorro_HH.itab_rev1 pbase/socorro.pbase 0 phreg0/socorro_HH.punw1 0 12901 phreg0/socorro_HH.pres2 phreg0/socorro_HH.pdh2 phreg0/socorro_HH.pddef2 phreg0/socorro_HH.punw2 phreg0/socorro_HH.psigma2 phreg0/socorro_HH.pmask2 30 - 0.01 20.0 1 - -
def_mod_pt plist/socorro_HV.plist phreg0/socorro_HV.pmask1 pslc/socorro_HH.pslc_par - tab/socorro_HV.itab_rev1 pbase/socorro.pbase 0 phreg0/socorro_HV.punw1 0 10554 phreg0/socorro_HV.pres2 phreg0/socorro_HV.pdh2 phreg0/socorro_HV.pddef2 phreg0/socorro_HV.punw2 phreg0/socorro_HV.psigma2 phreg0/socorro_HV.pmask2 30 - 0.01 20.0 1 - -
def_mod_pt plist/socorro_VV.plist phreg0/socorro_VV.pmask1 pslc/socorro_HH.pslc_par - tab/socorro_VV.itab_rev1 pbase/socorro.pbase 0 phreg0/socorro_VV.punw1 0 12783 phreg0/socorro_VV.pres2 phreg0/socorro_VV.pdh2 phreg0/socorro_VV.pddef2 phreg0/socorro_VV.punw2 phreg0/socorro_VV.psigma2 phreg0/socorro_VV.pmask2 30 - 0.01 20.0 1 - -
def_mod_pt plist/socorro_PCO.plist phreg0/socorro_PCO.pmask1 pslc/socorro_PCO.pslc_par - tab/socorro_PCO.itab_rev1 pbase/socorro_PCO.pbase 0 phreg0/socorro_PCO.punw1 0 33220 phreg0/socorro_PCO.pres2 phreg0/socorro_PCO.pdh2 phreg0/socorro_PCO.pddef2 phreg0/socorro_PCO.punw2 phreg0/socorro_PCO.psigma2 phreg0/socorro_PCO.pmask2 30 - 0.01 20.0 1 - -

# Creating residual phase rasters
prasdt_pwr24 plist/socorro_HH.plist phreg0/socorro_HH.pmask1 rslc_chip/20080802HH.rslc.par phreg0/socorro_HH.pres1 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_HV.plist phreg0/socorro_HV.pmask1 rslc_chip/20080802HH.rslc.par phreg0/socorro_HV.pres1 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_VV.plist phreg0/socorro_VV.pmask1 rslc_chip/20080802HH.rslc.par phreg0/socorro_VV.pres1 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_PCO.plist phreg0/socorro_PCO.pmask1 rslc_chip/20080802HH.rslc.par phreg0/socorro_PCO.pres1 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
```

pint1.sh

```
echo ... Calculating updated differential interferograms

## COMMAND REFERENCE
# lin_comb_pt <plist> <pmask> <pdata1>  <rec_num1> <pdata2> <rec_num2> <pdata3>  <rec_num3> <constant>
<factor1> <factor2>  <type> [pmask] [pt_num]
# phase_sim_pt <plist> <pmask> <pSLC_par>  <ppos> <itab> <rec_num> <pbase> <phgt>  <psim_unw> [pdef]
[ph_flag] [bflag]
# sub_phase_pt <plist> <pmask> <pint1>  <rec_num> <punw2> <pdiff> [int_type] [inverse]


# Updating phase model with point height corrections from regression analysis
lin_comb_pt plist/socorro_HH.plist phreg0/socorro_HH.pmask2 pint0/socorro_HH.pdem 1 phreg0/socorro_HH.pdh2 1
pint1/socorro_HH.phgt1 1 -0. 1. 1. 2 0
lin_comb_pt plist/socorro_HV.plist phreg0/socorro_HV.pmask2 pint0/socorro_HV.pdem 1 phreg0/socorro_HV.pdh2 1
pint1/socorro_HV.phgt1 1 -0. 1. 1. 2 0
lin_comb_pt plist/socorro_VV.plist phreg0/socorro_VV.pmask2 pint0/socorro_VV.pdem 1 phreg0/socorro_VV.pdh2 1
pint1/socorro_VV.phgt1 1 -0. 1. 1. 2 0
lin_comb_pt plist/socorro_PCO.plist phreg0/socorro_PCO.pmask2 pint0/socorro_PCO.pdem 1
phreg0/socorro_PCO.pdh2 1 pint1/socorro_PCO.phgt1 1 -0. 1. 1. 2 0

# Using improved point heights to re-simulate interferometric phase
phase_sim_pt plist/socorro_HH.plist phreg0/socorro_HH.pmask2 pslc/socorro_HH.pslc_par - tab/socorro_HH.itab -
pbase/socorro_HH.pbase pint1/socorro_HH.phgt1 pint1/socorro_HH.psim_unw0 - 0 0
phase_sim_pt plist/socorro_HV.plist phreg0/socorro_HV.pmask2 pslc/socorro_HH.pslc_par - tab/socorro_HV.itab -
pbase/socorro_HV.pbase pint1/socorro_HV.phgt1 pint1/socorro_HV.psim_unw0 - 0 0
phase_sim_pt plist/socorro_VV.plist phreg0/socorro_VV.pmask2 pslc/socorro_HH.pslc_par - tab/socorro_VV.itab -
pbase/socorro_VV.pbase pint1/socorro_VV.phgt1 pint1/socorro_VV.psim_unw0 - 0 0
phase_sim_pt plist/socorro_PCO.plist phreg0/socorro_PCO.pmask2 pslc/socorro_PCO.pslc_par -
tab/socorro_PCO.itab - pbase/socorro_PCO.pbase pint1/socorro_PCO.phgt1 pint1/socorro_PCO.psim_unw0 - 0 0

# Subtracting improved simulated phase model from the complex interferograms
sub_phase_pt plist/socorro_HH.plist phreg0/socorro_HH.pmask2 pint0/socorro_HH.pint -
pint1/socorro_HH.psim_unw0 pint1/socorro_HH.pdiff0 1 0
sub_phase_pt plist/socorro_HV.plist phreg0/socorro_HV.pmask2 pint0/socorro_HV.pint -
pint1/socorro_HV.psim_unw0 pint1/socorro_HV.pdiff0 1 0
sub_phase_pt plist/socorro_VV.plist phreg0/socorro_VV.pmask2 pint0/socorro_VV.pint -
pint1/socorro_VV.psim_unw0 pint1/socorro_VV.pdiff0 1 0
sub_phase_pt plist/socorro_PCO.plist phreg0/socorro_PCO.pmask2 pint0/socorro_PCO.pint -
pint1/socorro_PCO.psim_unw0 pint1/socorro_PCO.pdiff0 1 0

# Generating raster images of the updated differential interferograms
```

prasmph_pwr24 plist/socorro_HH.plist phreg0/socorro_HH.pmask2 rslc_chip/20080802HH.rslc.par pint1/socorro_HH.pdiff0 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 3
prasmph_pwr24 plist/socorro_HV.plist phreg0/socorro_HV.pmask2 rslc_chip/20080802HH.rslc.par pint1/socorro_HV.pdiff0 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 3
prasmph_pwr24 plist/socorro_VV.plist phreg0/socorro_VV.pmask2 rslc_chip/20080802HH.rslc.par pint1/socorro_VV.pdiff0 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 3
prasmph_pwr24 plist/socorro_PCO.plist phreg0/socorro_PCO.pmask2 rslc_chip/20080802HH.rslc.par pint1/socorro_PCO.pdiff0 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 3

base_ref.sh

echo ... performing baseline refinement

COMMAND REFERENCE
prt_pt <plist> <pdata> <np_start> <np> <type> [output] [rec_num] [n_rec]
lin_comb_pt <plist> <pmask> <pdata1> <rec_num1> <pdata2> <rec_num2> <pdata3> <rec_num3> <constant> <factor1> <factor2> <type> [pmask] [pt_num]
base_ls_pt <plist> <pmask> <pSLC_par> <ppos> <itab> <rec_num> <punw> <phgt> <pbase> [ph_flag] [bc_flag] [bn_flag] [bcdot_flag] [bndot_flag] [bperp_min]

Calculating unwrapped phase of initial interferograms by adding back what was initially subtracted
sub_phase_pt plist/socorro_HH.plist phreg0/socorro_HH.pmask2 pint0/socorro_HH.punw0 - pint0/socorro_HH.psim_unw0 base_ref/socorro_HH.pint_unw 0 1
sub_phase_pt plist/socorro_HV.plist phreg0/socorro_HV.pmask2 pint0/socorro_HV.punw0 - pint0/socorro_HV.psim_unw0 base_ref/socorro_HV.pint_unw 0 1
sub_phase_pt plist/socorro_VV.plist phreg0/socorro_VV.pmask2 pint0/socorro_VV.punw0 - pint0/socorro_VV.psim_unw0 base_ref/socorro_VV.pint_unw 0 1
sub_phase_pt plist/socorro_PCO.plist phreg0/socorro_PCO.pmask2 pint0/socorro_PCO.punw0 - pint0/socorro_PCO.psim_unw0 base_ref/socorro_PCO.pint_unw 0 1

Backing-up original baseline data and performing baseline refinement
cp pbase/socorro_HH.pbase base_ref/socorro_HH.pbase
cp pbase/socorro_HV.pbase base_ref/socorro_HV.pbase
cp pbase/socorro_VV.pbase base_ref/socorro_VV.pbase
cp pbase/socorro_PCO.pbase base_ref/socorro_PCO.pbase

base_ls_pt plist/socorro_HH.plist phreg0/socorro_HH.pmask2 pslc/socorro_HH.pslc_par - tab/socorro_HH.itab - base_ref/socorro_HH.pint_unw pint0/socorro_HH.pdem base_ref/socorro_HH.pbase 0 1 1 1 1 1. > base_ref/socorro_HH_base_ls_pt.out
base_ls_pt plist/socorro_HV.plist phreg0/socorro_HV.pmask2 pslc/socorro_HH.pslc_par - tab/socorro_HV.itab - base_ref/socorro_HV.pint_unw pint0/socorro_HV.pdem base_ref/socorro_HV.pbase 0 1 1 1 1 1. > base_ref/socorro_HV_base_ls_pt.out
base_ls_pt plist/socorro_VV.plist phreg0/socorro_VV.pmask2 pslc/socorro_HH.pslc_par - tab/socorro_VV.itab - base_ref/socorro_VV.pint_unw pint0/socorro_VV.pdem base_ref/socorro_VV.pbase 0 1 1 1 1 1. > base_ref/socorro_VV_base_ls_pt.out
base_ls_pt plist/socorro_PCO.plist phreg0/socorro_PCO.pmask2 pslc/socorro_PCO.pslc_par - tab/socorro_PCO.itab - base_ref/socorro_PCO.pint_unw pint0/socorro_PCO.pdem base_ref/socorro_PCO.pbase 0 1 1 1 1 1. > base_ref/socorro_PCO_base_ls_pt.out

```
grep "b_perp (t=center) (m):" base_ref/socorro_HH_base_ls_pt.out
grep "RMS altitude error (m):" base_ref/socorro_HH_base_ls_pt.out

# Calculating simulated phase using refined baselines
phase_sim_pt plist/socorro_HH.plist phreg0/socorro_HH.pmask2 pslc/socorro_HH.pslc_par - tab/socorro_HH.itab -
base_ref/socorro_HH.pbase pint1/socorro_HH.phgt1 base_ref/socorro_HH.psim_unw0 - 0 1
phase_sim_pt plist/socorro_HV.plist phreg0/socorro_HV.pmask2 pslc/socorro_HV.pslc_par - tab/socorro_HV.itab -
base_ref/socorro_HV.pbase pint1/socorro_HV.phgt1 base_ref/socorro_HV.psim_unw0 - 0 1
phase_sim_pt plist/socorro_VV.plist phreg0/socorro_VV.pmask2 pslc/socorro_VV.pslc_par - tab/socorro_VV.itab -
base_ref/socorro_VV.pbase pint1/socorro_VV.phgt1 base_ref/socorro_VV.psim_unw0 - 0 1
phase_sim_pt plist/socorro_PCO.plist phreg0/socorro_PCO.pmask2 pslc/socorro_PCO.pslc_par -
tab/socorro_PCO.itab - base_ref/socorro_PCO.pbase pint1/socorro_PCO.phgt1 base_ref/socorro_PCO.psim_unw0 - 0
1

# Subtracts refined simulated phase model from the unwrapped interferometric phase
sub_phase_pt plist/socorro_HH.plist phreg0/socorro_HH.pmask2 base_ref/socorro_HH.pint_unw -
base_ref/socorro_HH.psim_unw0 base_ref/socorro_HH.pdiff0_unw 0 0
sub_phase_pt plist/socorro_HV.plist phreg0/socorro_HV.pmask2 base_ref/socorro_HV.pint_unw -
base_ref/socorro_HV.psim_unw0 base_ref/socorro_HV.pdiff0_unw 0 0
sub_phase_pt plist/socorro_VV.plist phreg0/socorro_VV.pmask2 base_ref/socorro_VV.pint_unw -
base_ref/socorro_VV.psim_unw0 base_ref/socorro_VV.pdiff0_unw 0 0
sub_phase_pt plist/socorro_PCO.plist phreg0/socorro_PCO.pmask2 base_ref/socorro_PCO.pint_unw -
base_ref/socorro_PCO.psim_unw0 base_ref/socorro_PCO.pdiff0_unw 0 0

# Re-running regresssion analysis with refined baselines
def_mod_pt plist/socorro_HH.plist phreg0/socorro_HH.pmask2 pslc/socorro_HH.pslc_par - tab/socorro_HH.itab
base_ref/socorro_HH.pbase 1 base_ref/socorro_HH.pdiff0_unw 0 12901 base_ref/socorro_HH.pres1
base_ref/socorro_HH.pdh1 - - base_ref/socorro_HH.psigma1 base_ref/socorro_HH.pmask1 30 -0.03 0.03 3.0 1 - -
def_mod_pt plist/socorro_HV.plist phreg0/socorro_HV.pmask2 pslc/socorro_HH.pslc_par - tab/socorro_HV.itab
base_ref/socorro_HV.pbase 1 base_ref/socorro_HV.pdiff0_unw 0 10554 base_ref/socorro_HV.pres1
base_ref/socorro_HV.pdh1 - - base_ref/socorro_HV.psigma1 base_ref/socorro_HV.pmask1 30 -0.03 0.03 3.0 1 - -
def_mod_pt plist/socorro_VV.plist phreg0/socorro_VV.pmask2 pslc/socorro_HH.pslc_par - tab/socorro_VV.itab
base_ref/socorro_VV.pbase 1 base_ref/socorro_VV.pdiff0_unw 0 12783 base_ref/socorro_VV.pres1
base_ref/socorro_VV.pdh1 - - base_ref/socorro_VV.psigma1 base_ref/socorro_VV.pmask1 30 -0.03 0.03 3.0 1 - -
def_mod_pt plist/socorro_PCO.plist phreg0/socorro_PCO.pmask2 pslc/socorro_PCO.pslc_par - tab/socorro_PCO.itab
base_ref/socorro_PCO.pbase 1 base_ref/socorro_PCO.pdiff0_unw 0 33220 base_ref/socorro_PCO.pres1
base_ref/socorro_PCO.pdh1 - - base_ref/socorro_PCO.psigma1 base_ref/socorro_PCO.pmask1 30 -0.03 0.03 3.0 1 - -

# Creating output rasters
prasdt_pwr24 plist/socorro_HH.plist base_ref/socorro_HH.pmask1 rslc_chip/20080802HH.rslc.par
base_ref/socorro_HH.pres1 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_HV.plist base_ref/socorro_HV.pmask1 rslc_chip/20080802HH.rslc.par
base_ref/socorro_HV.pres1 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_VV.plist base_ref/socorro_VV.pmask1 rslc_chip/20080802HH.rslc.par
base_ref/socorro_VV.pres1 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_PCO.plist base_ref/socorro_PCO.pmask1 rslc_chip/20080802HH.rslc.par
base_ref/socorro_PCO.pres1 - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3

prasdt_pwr24 plist/socorro_HH.plist base_ref/socorro_HH.pmask1 rslc_chip/20080802HH.rslc.par
base_ref/socorro_HH.pdiff0_unw - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_HV.plist base_ref/socorro_HV.pmask1 rslc_chip/20080802HH.rslc.par
base_ref/socorro_HV.pdiff0_unw - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_VV.plist base_ref/socorro_VV.pmask1 rslc_chip/20080802HH.rslc.par
base_ref/socorro_VV.pdiff0_unw - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_PCO.plist base_ref/socorro_PCO.pmask1 rslc_chip/20080802HH.rslc.par
base_ref/socorro_PCO.pdiff0_unw - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
```

pint2.sh

```
echo ... calculating updated unwrapped differential interferograms

## COMMAND REFERENCE ##
# cct_pt <plist> <pmask> <SLC_par> <pdata_in>  <pcct> [type] [r_max] [w_func] [np_min]


# Updating point height map
lin_comb_pt plist/socorro_HH.plist base_ref/socorro_HH.pmask1 pint1/socorro_HH.phgt1 1
base_ref/socorro_HH.pdh1 1 pint2/socorro_HH.phgt2 1 -0. 1. 1. 2 0
lin_comb_pt plist/socorro_HV.plist base_ref/socorro_HV.pmask1 pint1/socorro_HV.phgt1 1
base_ref/socorro_HV.pdh1 1 pint2/socorro_HV.phgt2 1 -0. 1. 1. 2 0
lin_comb_pt plist/socorro_VV.plist base_ref/socorro_VV.pmask1 pint1/socorro_VV.phgt1 1
base_ref/socorro_VV.pdh1 1 pint2/socorro_VV.phgt2 1 -0. 1. 1. 2 0
lin_comb_pt plist/socorro_PCO.plist base_ref/socorro_PCO.pmask1 pint1/socorro_PCO.phgt1 1
base_ref/socorro_PCO.pdh1 1 pint2/socorro_PCO.phgt2 1 -0. 1. 1. 2 0

# Re-calculating the simulated phase model using improved point heights from the refined regression analysis
phase_sim_pt plist/socorro_HH.plist base_ref/socorro_HH.pmask1 pslc/socorro_HH.pslc_par - tab/socorro_HH.itab -
base_ref/socorro_HH.pbase pint2/socorro_HH.phgt2 pint2/socorro_HH.psim_unw1 - 0 1
phase_sim_pt plist/socorro_HV.plist base_ref/socorro_HV.pmask1 pslc/socorro_HH.pslc_par - tab/socorro_HV.itab -
base_ref/socorro_HV.pbase pint2/socorro_HV.phgt2 pint2/socorro_HV.psim_unw1 - 0 1
phase_sim_pt plist/socorro_VV.plist base_ref/socorro_VV.pmask1 pslc/socorro_HH.pslc_par - tab/socorro_VV.itab -
base_ref/socorro_VV.pbase pint2/socorro_VV.phgt2 pint2/socorro_VV.psim_unw1 - 0 1
phase_sim_pt plist/socorro_PCO.plist base_ref/socorro_PCO.pmask1 pslc/socorro_PCO.pslc_par -
tab/socorro_PCO.itab - base_ref/socorro_PCO.pbase pint2/socorro_PCO.phgt2 pint2/socorro_PCO.psim_unw1 - 0 1

# Subtracting improved phase model
sub_phase_pt plist/socorro_HH.plist base_ref/socorro_HH.pmask1 base_ref/socorro_HH.pint_unw -
pint2/socorro_HH.psim_unw1 pint2/socorro_HH.pdiff1_unw 0 0
sub_phase_pt plist/socorro_HV.plist base_ref/socorro_HV.pmask1 base_ref/socorro_HV.pint_unw -
pint2/socorro_HV.psim_unw1 pint2/socorro_HV.pdiff1_unw 0 0
sub_phase_pt plist/socorro_VV.plist base_ref/socorro_VV.pmask1 base_ref/socorro_VV.pint_unw -
pint2/socorro_VV.psim_unw1 pint2/socorro_VV.pdiff1_unw 0 0
sub_phase_pt plist/socorro_PCO.plist base_ref/socorro_PCO.pmask1 base_ref/socorro_PCO.pint_unw -
pint2/socorro_PCO.psim_unw1 pint2/socorro_PCO.pdiff1_unw 0 0
```

```
# Calculate temporal coherence for point quality measure
cct_pt plist/socorro_HH.plist base_ref/socorro_HH.pmask1 rslc_chip/20080802HH.rslc.par
base_ref/socorro_HH.pdiff0_unw pint2/socorro_HH.pcct 2
cct_pt plist/socorro_HV.plist base_ref/socorro_HV.pmask1 rslc_chip/20080802HH.rslc.par
base_ref/socorro_HV.pdiff0_unw pint2/socorro_HV.pcct 2
cct_pt plist/socorro_VV.plist base_ref/socorro_VV.pmask1 rslc_chip/20080802HH.rslc.par
base_ref/socorro_VV.pdiff0_unw pint2/socorro_VV.pcct 2
cct_pt plist/socorro_PCO.plist base_ref/socorro_PCO.pmask1 rslc_chip/20080802HH.rslc.par
base_ref/socorro_PCO.pdiff0_unw pint2/socorro_PCO.pcct 2

# Generating output rasters of the refined unwrapped differential interferograms and temporal coherence maps
prasdt_pwr24 plist/socorro_HH.plist base_ref/socorro_HH.pmask1 rslc_chip/20080802HH.rslc.par
pint2/socorro_HH.pdiff1_unw - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_HV.plist base_ref/socorro_HV.pmask1 rslc_chip/20080802HH.rslc.par
pint2/socorro_HV.pdiff1_unw - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_VV.plist base_ref/socorro_VV.pmask1 rslc_chip/20080802HH.rslc.par
pint2/socorro_VV.pdiff1_unw - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3
prasdt_pwr24 plist/socorro_PCO.plist base_ref/socorro_PCO.pmask1 rslc_chip/20080802HH.rslc.par
pint2/socorro_PCO.pdiff1_unw - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 6.28 3

prasdt_pwr24 plist/socorro_HH.plist base_ref/socorro_HH.pmask1 rslc_chip/20080802HH.rslc.par
pint2/socorro_HH.pcct - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 1.0 3
prasdt_pwr24 plist/socorro_HV.plist base_ref/socorro_HV.pmask1 rslc_chip/20080802HH.rslc.par
pint2/socorro_HV.pcct - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 1.0 3
prasdt_pwr24 plist/socorro_VV.plist base_ref/socorro_VV.pmask1 rslc_chip/20080802HH.rslc.par
pint2/socorro_VV.pcct - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 1.0 3
prasdt_pwr24 plist/socorro_PCO.plist base_ref/socorro_PCO.pmask1 rslc_chip/20080802HH.rslc.par
pint2/socorro_PCO.pcct - rslc_chip/20080802HH.rslc.par pwr/20080802HH.pwr 1.0 3
```

7.7. Appendix G: MATLAB Processing Script

cs_analysis.m

```
%% POLARIMETRIC SCATTERER ANALYSIS

clear

root_dir = 'C:\Documents and Settings\garner\My Documents\jakes_work\ipta3\';
root_dir2 = 'C:\work\SARscape\socorro\20100206\';

%% Generate power image
slc_name = [root_dir  'rslc_chip\20080802HH.rslc'];
par_name = [root_dir  'rslc_chip\20080802HH.rslc.par'];

[X, dims] = read_gamma_slc( slc_name, par_name );
X = 10*log10(X.*conj(X));


%% read gamma point lists and masks
HH_plist = read_gamma_plist([root_dir 'plist\socorro_HH.plist']);
HV_plist = read_gamma_plist([root_dir 'plist\socorro_HV.plist']);
VV_plist = read_gamma_plist([root_dir 'plist\socorro_VV.plist']);
PCO_plist = read_gamma_plist([root_dir 'plist\socorro_PCO.plist']);

HH_pmask = read_gamma_pmask([root_dir 'base_ref\socorro_HH.pmask1']);
HV_pmask = read_gamma_pmask([root_dir 'base_ref\socorro_HV.pmask1']);
VV_pmask = read_gamma_pmask([root_dir 'base_ref\socorro_VV.pmask1']);
PCO_pmask = read_gamma_pmask([root_dir 'base_ref\socorro_PCO.pmask1']);

% shift coordinate origin for (0,0) to (1,1)
HH_plist = [HH_plist(:,1)+1, HH_plist(:,2)+1];
HV_plist = [HV_plist(:,1)+1, HV_plist(:,2)+1];
VV_plist = [VV_plist(:,1)+1, VV_plist(:,2)+1];
PCO_plist = [PCO_plist(:,1)+1, PCO_plist(:,2)+1];

%% Plot initial plists
```

```
figure(1); imagesc(X ,[0 100]);
colormap gray; hold on;
title('Initial Point List Locations: HH');
plot(HH_plist(:,1),HH_plist(:,2),'r.','MarkerSize',3);
hold off;

figure(2); imagesc(X ,[0 100]);
colormap gray; hold on;
title('Initial Point List Locations: HV');
plot(HV_plist(:,1),HV_plist(:,2),'r.','MarkerSize',3);
hold off;

figure(3); imagesc(X ,[0 100]);
colormap gray; hold on;
title('Initial Point List Locations: VV');
plot(VV_plist(:,1),VV_plist(:,2),'r.','MarkerSize',3);
hold off;

figure(4); imagesc(X ,[0 100]);
colormap gray; hold on;
title('Initial Point List Locations: PCO');
plot(PCO_plist(:,1),PCO_plist(:,2),'r.','MarkerSize',3);
hold off;


%% read temporal coherence (should match # of pts)
HH_pcct = read_gamma_pdata([root_dir 'pint2\socorro_HH.pcct']);
HV_pcct = read_gamma_pdata([root_dir 'pint2\socorro_HV.pcct']);
VV_pcct = read_gamma_pdata([root_dir 'pint2\socorro_VV.pcct']);
PCO_pcct = read_gamma_pdata([root_dir 'pint2\socorro_PCO.pcct']);

%% remove zeros present from plist masking
HH_plist = HH_plist(HH_pmask == 1,:);
HV_plist = HV_plist(HV_pmask == 1,:);
VV_plist = VV_plist(VV_pmask == 1,:);
PCO_plist = PCO_plist(PCO_pmask == 1,:);

HH_pcct = HH_pcct(HH_pmask == 1);
HV_pcct = HV_pcct(HV_pmask == 1);
VV_pcct = VV_pcct(VV_pmask == 1);
PCO_pcct = PCO_pcct(PCO_pmask == 1);

%% Plot masked plists

figure(1); %imagesc(X ,[0 100]);
colormap gray; hold on;
title('Final Point List Locations: HH');
plot(HH_plist(:,1),HH_plist(:,2),'c.','MarkerSize',3);
hold off;

figure(2); %imagesc(X ,[0 100]);
colormap gray; hold on;
title('Final Point List Locations: HV');
plot(HV_plist(:,1),HV_plist(:,2),'c.','MarkerSize',3);
hold off;

figure(3); %imagesc(X ,[0 100]);
colormap gray; hold on;
title('Final Point List Locations: VV');
```

```
plot(VV_plist(:,1),VV_plist(:,2),'c.','MarkerSize',3);
hold off;

figure(4); %imagesc(X ,[0 100]);
colormap gray; hold on;
title('Final Point List Locations: PCO');
plot(PCO_plist(:,1),PCO_plist(:,2),'c.','MarkerSize',3);
hold off;

%% Point List Comparison

figure(5); imagesc(X ,[0 100]);
colormap gray; hold on;
title('Final Point List Locations: PCO');
plot(PCO_plist(:,1),PCO_plist(:,2),'r.','MarkerSize',3);
plot(HH_plist(:,1),HH_plist(:,2),'c.','MarkerSize',3);
hold off;

figure(6); imagesc(X ,[0 100]);
colormap gray; hold on;
title('Final Point List Locations: PCO');
plot(PCO_plist(:,1),PCO_plist(:,2),'r.','MarkerSize',3);
plot(HV_plist(:,1),HV_plist(:,2),'c.','MarkerSize',3);
hold off;

figure(7); imagesc(X ,[0 100]);
colormap gray; hold on;
title('Final Point List Locations: PCO');
plot(PCO_plist(:,1),PCO_plist(:,2),'r.','MarkerSize',3);
plot(VV_plist(:,1),VV_plist(:,2),'c.','MarkerSize',3);
hold off;


%% Calculate common points plists
HH_plist_mat = zeros(1200,1200,'uint8');
HV_plist_mat = zeros(1200,1200,'uint8');
VV_plist_mat = zeros(1200,1200,'uint8');
PCO_plist_mat = zeros(1200,1200,'uint8');

for n = 1:size(HH_plist,1)
   HH_plist_mat(HH_plist(n,2),HH_plist(n,1)) = 1;
end

for n = 1:size(HV_plist,1)
   HV_plist_mat(HV_plist(n,2),HV_plist(n,1)) = 1;
end

for n = 1:size(VV_plist,1)
   VV_plist_mat(VV_plist(n,2),VV_plist(n,1)) = 1;
end

for n = 1:size(PCO_plist,1)
   PCO_plist_mat(PCO_plist(n,2),PCO_plist(n,1)) = 1;
end


PCO_HH_common_mat = HH_plist_mat .* PCO_plist_mat;
[PCO_HH_common_plist(:,2) PCO_HH_common_plist(:,1)] = find(PCO_HH_common_mat==1);

PCO_HV_common_mat = HV_plist_mat .* PCO_plist_mat;
```

```
[PCO_HV_common_plist(:,2) PCO_HV_common_plist(:,1)] = find(PCO_HV_common_mat==1);

PCO_VV_common_mat = VV_plist_mat .* PCO_plist_mat;
[PCO_VV_common_plist(:,2) PCO_VV_common_plist(:,1)] = find(PCO_VV_common_mat==1);

HH_HV_common_mat = HH_plist_mat .* HV_plist_mat;
[HH_HV_common_plist(:,2) HH_HV_common_plist(:,1)] = find(HH_HV_common_mat==1);

HH_VV_common_mat = HH_plist_mat .* VV_plist_mat;
[HH_VV_common_plist(:,2) HH_VV_common_plist(:,1)] = find(HH_VV_common_mat==1);

VV_HV_common_mat = VV_plist_mat .* HV_plist_mat;
[VV_HV_common_plist(:,2) VV_HV_common_plist(:,1)] = find(VV_HV_common_mat==1);



% point density
pdensity_mat = HH_plist_mat + HV_plist_mat + VV_plist_mat;% + PCO_plist_mat;

[pden_1_plist(:,2) pden_1_plist(:,1)] = find(pdensity_mat==1);
[pden_2_plist(:,2) pden_2_plist(:,1)] = find(pdensity_mat==2);
[pden_3_plist(:,2) pden_3_plist(:,1)] = find(pdensity_mat==3);

figure(7); imagesc(X ,[0 100]);
colormap gray; hold on;
title('Point Target Density');
plot(pden_1_plist(:,1),pden_1_plist(:,2),'b.','MarkerSize',3);
plot(pden_2_plist(:,1),pden_2_plist(:,2),'r.','MarkerSize',3);
plot(pden_3_plist(:,1),pden_3_plist(:,2),'y.','MarkerSize',3);
hold off;

figure; imagesc(X ,[0 100]);
colormap gray; hold on;
title('Common Point Targets between HH and PCO Point Lists');
plot(PCO_HH_common_plist(:,1),PCO_HH_common_plist(:,2),'r.','MarkerSize',3);
hold off;

figure; imagesc(X ,[0 100]);
colormap gray; hold on;
title('Common Point Targets between HV and PCO Point Lists');
plot(PCO_HV_common_plist(:,1),PCO_HV_common_plist(:,2),'r.','MarkerSize',3);
hold off;

figure; imagesc(X ,[0 100]);
colormap gray; hold on;
title('Common Point Targets between VV and PCO Point Lists');
plot(PCO_VV_common_plist(:,1),PCO_VV_common_plist(:,2),'r.','MarkerSize',3);
hold off;

figure; imagesc(X ,[0 100]);
colormap gray; hold on;
title('Common Point Targets between HH and HV Point Lists');
plot(HH_HV_common_plist(:,1),HH_HV_common_plist(:,2),'r.','MarkerSize',3);
hold off;

figure; imagesc(X ,[0 100]);
colormap gray; hold on;
title('Common Point Targets between HH and VV Point Lists');
plot(HH_VV_common_plist(:,1),HH_VV_common_plist(:,2),'r.','MarkerSize',3);
hold off;
```

```
figure; imagesc(X ,[0 100]);
colormap gray; hold on;
title('Common Point Targets between VV and HV Point Lists');
plot(VV_HV_common_plist(:,1),VV_HV_common_plist(:,2),'r.','MarkerSize',3);
hold off;
%% remove zeros present from cct calculation
HH_plist = HH_plist(HH_pcct ~= 0,:);
HV_plist = HV_plist(HV_pcct ~= 0,:);
VV_plist = VV_plist(VV_pcct ~= 0,:);
PCO_plist = PCO_plist(PCO_pcct ~= 0,:);

HH_pcct = HH_pcct(HH_pcct ~= 0);
HV_pcct = HV_pcct(HV_pcct ~= 0);
VV_pcct = VV_pcct(VV_pcct ~= 0);
PCO_pcct = PCO_pcct(PCO_pcct ~= 0);


%% calculate cct histograms and distributions stats for each pol

figure;
subplot(2,2,1); hist(HH_pcct,100);
subplot(2,2,2); hist(HV_pcct,100);
subplot(2,2,3); hist(VV_pcct,100);
subplot(2,2,4); hist(PCO_pcct,100);

mean(HH_pcct)
mean(HV_pcct)
mean(VV_pcct)
mean(PCO_pcct)

std(HH_pcct)
std(HV_pcct)
std(VV_pcct)
std(PCO_pcct)

var(HH_pcct)
var(HV_pcct)
var(VV_pcct)
var(PCO_pcct)


%% read in subsetted decomp images

% extract Pauli decomp point data with pts coords
pauli_K1 = read_envi_data([root_dir2 'pauli\sub_20080802_pauli_K1_slc_msb'],6,[1200 1200]);
pauli_K2 = read_envi_data([root_dir2 'pauli\sub_20080802_pauli_K2_slc_msb'],6,[1200 1200]);
pauli_K3 = read_envi_data([root_dir2 'pauli\sub_20080802_pauli_K3_slc_msb'],6,[1200 1200]);

% imagesc(abs(pauli_K1));

%% extract point data from Pauli images
for n = 1:size(HH_plist,1)
   HH_pauli_K1(n,1) = pauli_K1(HH_plist(n,2), HH_plist(n,1));
   HH_pauli_K2(n,1) = pauli_K2(HH_plist(n,2), HH_plist(n,1));
   HH_pauli_K3(n,1) = pauli_K3(HH_plist(n,2), HH_plist(n,1));
end

for n = 1:size(HV_plist,1)
   HV_pauli_K1(n,1) = pauli_K1(HV_plist(n,2), HV_plist(n,1));
```

```
    HV_pauli_K2(n,1) = pauli_K2(HV_plist(n,2), HV_plist(n,1));
    HV_pauli_K3(n,1) = pauli_K3(HV_plist(n,2), HV_plist(n,1));
end

for n = 1:size(VV_plist,1)
    VV_pauli_K1(n,1) = pauli_K1(VV_plist(n,2), VV_plist(n,1));
    VV_pauli_K2(n,1) = pauli_K2(VV_plist(n,2), VV_plist(n,1));
    VV_pauli_K3(n,1) = pauli_K3(VV_plist(n,2), VV_plist(n,1));
end

for n = 1:size(PCO_plist,1)
    PCO_pauli_K1(n,1) = pauli_K1(PCO_plist(n,2), PCO_plist(n,1));
    PCO_pauli_K2(n,1) = pauli_K2(PCO_plist(n,2), PCO_plist(n,1));
    PCO_pauli_K3(n,1) = pauli_K3(PCO_plist(n,2), PCO_plist(n,1));
end

clear pauli_K1 pauli_K2 pauli_K3;

%% perform Pauli classification
HH_pauli_class = zeros([size(HH_plist,1) 1],'int8');
HH_pauli_class( (HH_pauli_K1 >= HH_pauli_K2) & (HH_pauli_K1 >= HH_pauli_K3) ) = 1;
HH_pauli_class( (HH_pauli_K2 >= HH_pauli_K1) & (HH_pauli_K2 >= HH_pauli_K3) ) = 2;
HH_pauli_class( (HH_pauli_K3 >= HH_pauli_K1) & (HH_pauli_K3 >= HH_pauli_K2) ) = 3;

HV_pauli_class = zeros([size(HV_plist,1) 1],'int8');
HV_pauli_class( (HV_pauli_K1 >= HV_pauli_K2) & (HV_pauli_K1 >= HV_pauli_K3) ) = 1;
HV_pauli_class( (HV_pauli_K2 >= HV_pauli_K1) & (HV_pauli_K2 >= HV_pauli_K3) ) = 2;
HV_pauli_class( (HV_pauli_K3 >= HV_pauli_K1) & (HV_pauli_K3 >= HV_pauli_K2) ) = 3;

VV_pauli_class = zeros([size(VV_plist,1) 1],'int8');
VV_pauli_class( (VV_pauli_K1 >= VV_pauli_K2) & (VV_pauli_K1 >= VV_pauli_K3) ) = 1;
VV_pauli_class( (VV_pauli_K2 >= VV_pauli_K1) & (VV_pauli_K2 >= VV_pauli_K3) ) = 2;
VV_pauli_class( (VV_pauli_K3 >= VV_pauli_K1) & (VV_pauli_K3 >= VV_pauli_K2) ) = 3;

PCO_pauli_class = zeros([size(PCO_plist,1) 1],'int8');
PCO_pauli_class( (PCO_pauli_K1 >= PCO_pauli_K2) & (PCO_pauli_K1 >= PCO_pauli_K3) ) = 1;
PCO_pauli_class( (PCO_pauli_K2 >= PCO_pauli_K1) & (PCO_pauli_K2 >= PCO_pauli_K3) ) = 2;
PCO_pauli_class( (PCO_pauli_K3 >= PCO_pauli_K1) & (PCO_pauli_K3 >= PCO_pauli_K2) ) = 3;

%% Pauli first order statisitics

HH_pauli_class_K1 = HH_pauli_class(HH_pauli_class==1);
HH_pauli_class_K2 = HH_pauli_class(HH_pauli_class==2);
HH_pauli_class_K3 = HH_pauli_class(HH_pauli_class==3);

disp(['Pauli HH K1 |' '| num pts: ' num2str(size(HH_pauli_class_K1,1))...
    '| mean: ' num2str(mean( abs(HH_pauli_K1(HH_pauli_class==1))))...
    '| std: ' num2str(std( abs(HH_pauli_K1(HH_pauli_class==1))))...
    '| var: ' num2str(var( abs(HH_pauli_K1(HH_pauli_class==1)))) ]);
disp(['Pauli HH K2 |' '| num pts: ' num2str(size(HH_pauli_class_K2,1))...
    '| mean: ' num2str(mean( abs(HH_pauli_K2(HH_pauli_class==2))))...
    '| std: ' num2str(std( abs(HH_pauli_K2(HH_pauli_class==2))))...
    '| var: ' num2str(var( abs(HH_pauli_K2(HH_pauli_class==2)))) ]);
disp(['Pauli HH K3 |' '| num pts: ' num2str(size(HH_pauli_class_K3,1))...
    '| mean: ' num2str(mean( abs(HH_pauli_K3(HH_pauli_class==3))))...
    '| std: ' num2str(std( abs(HH_pauli_K3(HH_pauli_class==3))))...
    '| var: ' num2str(var( abs(HH_pauli_K3(HH_pauli_class==3)))) ]);

HV_pauli_class_K1 = HV_pauli_class(HV_pauli_class==1);
HV_pauli_class_K2 = HV_pauli_class(HV_pauli_class==2);
```

```
HV_pauli_class_K3 = HV_pauli_class(HV_pauli_class==3);

disp(['Pauli HV K1 |' '| num pts: ' num2str(size(HV_pauli_class_K1,1))...
    '| mean: ' num2str(mean( abs(HV_pauli_K1(HV_pauli_class==1))))...
    '| std: ' num2str(std( abs(HV_pauli_K1(HV_pauli_class==1))))...
    '| var: ' num2str(var( abs(HV_pauli_K1(HV_pauli_class==1)))) ]);
disp(['Pauli HV K2 |' '| num pts: ' num2str(size(HV_pauli_class_K2,1))...
    '| mean: ' num2str(mean( abs(HV_pauli_K2(HV_pauli_class==2))))...
    '| std: ' num2str(std( abs(HV_pauli_K2(HV_pauli_class==2))))...
    '| var: ' num2str(var( abs(HV_pauli_K2(HV_pauli_class==2)))) ]);
disp(['Pauli HV K3 |' '| num pts: ' num2str(size(HV_pauli_class_K3,1))...
    '| mean: ' num2str(mean( abs(HV_pauli_K3(HV_pauli_class==3))))...
    '| std: ' num2str(std( abs(HV_pauli_K3(HV_pauli_class==3))))...
    '| var: ' num2str(var( abs(HV_pauli_K3(HV_pauli_class==3)))) ]);

VV_pauli_class_K1 = VV_pauli_class(VV_pauli_class==1);
VV_pauli_class_K2 = VV_pauli_class(VV_pauli_class==2);
VV_pauli_class_K3 = VV_pauli_class(VV_pauli_class==3);

disp(['Pauli VV K1 |' '| num pts: ' num2str(size(VV_pauli_class_K1,1))...
    '| mean: ' num2str(mean( abs(VV_pauli_K1(VV_pauli_class==1))))...
    '| std: ' num2str(std( abs(VV_pauli_K1(VV_pauli_class==1))))...
    '| var: ' num2str(var( abs(VV_pauli_K1(VV_pauli_class==1)))) ]);
disp(['Pauli VV K2 |' '| num pts: ' num2str(size(VV_pauli_class_K2,1))...
    '| mean: ' num2str(mean( abs(VV_pauli_K2(VV_pauli_class==2))))...
    '| std: ' num2str(std( abs(VV_pauli_K2(VV_pauli_class==2))))...
    '| var: ' num2str(var( abs(VV_pauli_K2(VV_pauli_class==2)))) ]);
disp(['Pauli VV K3 |' '| num pts: ' num2str(size(VV_pauli_class_K3,1))...
    '| mean: ' num2str(mean( abs(VV_pauli_K3(VV_pauli_class==3))))...
    '| std: ' num2str(std( abs(VV_pauli_K3(VV_pauli_class==3))))...
    '| var: ' num2str(var( abs(VV_pauli_K3(VV_pauli_class==3)))) ]);

PCO_pauli_class_K1 = PCO_pauli_class(PCO_pauli_class==1);
PCO_pauli_class_K2 = PCO_pauli_class(PCO_pauli_class==2);
PCO_pauli_class_K3 = PCO_pauli_class(PCO_pauli_class==3);

disp(['Pauli PCO K1 |' '| num pts: ' num2str(size(PCO_pauli_class_K1,1))...
    '| mean: ' num2str(mean( abs(PCO_pauli_K1(PCO_pauli_class==1))))...
    '| std: ' num2str(std( abs(PCO_pauli_K1(PCO_pauli_class==1))))...
    '| var: ' num2str(var( abs(PCO_pauli_K1(PCO_pauli_class==1)))) ]);
disp(['Pauli PCO K2 |' '| num pts: ' num2str(size(PCO_pauli_class_K2,1))...
    '| mean: ' num2str(mean( abs(PCO_pauli_K2(PCO_pauli_class==2))))...
    '| std: ' num2str(std( abs(PCO_pauli_K2(PCO_pauli_class==2))))...
    '| var: ' num2str(var( abs(PCO_pauli_K2(PCO_pauli_class==2)))) ]);
disp(['Pauli PCO K3 |' '| num pts: ' num2str(size(PCO_pauli_class_K3,1))...
    '| mean: ' num2str(mean( abs(PCO_pauli_K3(PCO_pauli_class==3))))...
    '| std: ' num2str(std( abs(PCO_pauli_K3(PCO_pauli_class==3))))...
    '| var: ' num2str(var( abs(PCO_pauli_K3(PCO_pauli_class==3)))) ]);



%% plot Pauli values vs. cct

[HH_bar(:,1) xout] = hist(HH_pcct((HH_pauli_class==1) & (HH_pcct > 0)),100);
[HH_bar(:,2) xout] = hist(HH_pcct((HH_pauli_class==2) & (HH_pcct > 0)),100);
[HH_bar(:,3) xout] = hist(HH_pcct((HH_pauli_class==3) & (HH_pcct > 0)),100);
figure; plot(xout,HH_bar);

[HV_bar(:,1) xout] = hist(HV_pcct((HV_pauli_class==1) & (HV_pcct > 0)),100);
[HV_bar(:,2) xout] = hist(HV_pcct((HV_pauli_class==2) & (HV_pcct > 0)),100);
```

```
[HV_bar(:,3) xout] = hist(HV_pcct((HV_pauli_class==3) & (HV_pcct > 0)),100);
figure; plot(xout,HV_bar);

[VV_bar(:,1) xout] = hist(VV_pcct((VV_pauli_class==1) & (VV_pcct > 0)),100);
[VV_bar(:,2) xout] = hist(VV_pcct((VV_pauli_class==2) & (VV_pcct > 0)),100);
[VV_bar(:,3) xout] = hist(VV_pcct((VV_pauli_class==3) & (VV_pcct > 0)),100);
figure; plot(xout,VV_bar);

[PCO_bar(:,1) xout] = hist(PCO_pcct((PCO_pauli_class==1) & (PCO_pcct > 0)),100);
[PCO_bar(:,2) xout] = hist(PCO_pcct((PCO_pauli_class==2) & (PCO_pcct > 0)),100);
[PCO_bar(:,3) xout] = hist(PCO_pcct((PCO_pauli_class==3) & (PCO_pcct > 0)),100);
figure; plot(xout,PCO_bar);

figure; scatterhist(HH_pcct,abs(HH_pauli_K1),[100 100]);
figure; scatterhist(HH_pcct,abs(HH_pauli_K2),[100 100]);
figure; scatterhist(HH_pcct,abs(HH_pauli_K3),[100 100]);

figure; scatterhist(PCO_pcct,abs(PCO_pauli_K1),[100 100]);
figure; scatterhist(PCO_pcct,abs(PCO_pauli_K2),[100 100]);
figure; scatterhist(PCO_pcct,abs(PCO_pauli_K3),[100 100]);



%% extract a-H decomp point data with pts coords

aH_class = read_envi_data([root_dir2 'a-H2\sub_20080802_aH_class'],1,[1200 1200]);
aH_alpha = read_envi_data([root_dir2 'a-H2\sub_20080802_aH_alpha_msb'],4,[1200 1200]);
aH_entropy = read_envi_data([root_dir2 'a-H2\sub_20080802_aH_entropy_msb'],4,[1200 1200]);

%% extract point data from a-H image in zones
for n = 1:size(HH_plist,1)
   HH_aH_class(n,1) = aH_class(HH_plist(n,2), HH_plist(n,1));
   HH_aH_alpha(n,1) = aH_alpha(HH_plist(n,2), HH_plist(n,1));
   HH_aH_entropy(n,1) = aH_entropy(HH_plist(n,2), HH_plist(n,1));
end

for n = 1:size(HV_plist,1)
   HV_aH_class(n,1) = aH_class(HV_plist(n,2), HV_plist(n,1));
   HV_aH_alpha(n,1) = aH_alpha(HV_plist(n,2), HV_plist(n,1));
   HV_aH_entropy(n,1) = aH_entropy(HV_plist(n,2), HV_plist(n,1));
end

for n = 1:size(VV_plist,1)
   VV_aH_class(n,1) = aH_class(VV_plist(n,2), VV_plist(n,1));
   VV_aH_alpha(n,1) = aH_alpha(VV_plist(n,2), VV_plist(n,1));
   VV_aH_entropy(n,1) = aH_entropy(VV_plist(n,2), VV_plist(n,1));
end

for n = 1:size(PCO_plist,1)
   PCO_aH_class(n,1) = aH_class(PCO_plist(n,2), PCO_plist(n,1));
   PCO_aH_alpha(n,1) = aH_alpha(PCO_plist(n,2), PCO_plist(n,1));
   PCO_aH_entropy(n,1) = aH_entropy(PCO_plist(n,2), PCO_plist(n,1));
end

% extracting class values
HH_aH_class_Z1 = HH_aH_class(HH_aH_class==2);
HH_aH_class_Z2 = HH_aH_class(HH_aH_class==1);
HH_aH_class_Z3 = HH_aH_class(HH_aH_class==0);
HH_aH_class_Z4 = HH_aH_class(HH_aH_class==5);
HH_aH_class_Z5 = HH_aH_class(HH_aH_class==4);
```

```
HH_aH_class_Z6 = HH_aH_class(HH_aH_class==3);
HH_aH_class_Z7 = HH_aH_class(HH_aH_class==8);
HH_aH_class_Z8 = HH_aH_class(HH_aH_class==7);
HH_aH_class_Z9 = HH_aH_class(HH_aH_class==6);

HV_aH_class_Z1 = HV_aH_class(HV_aH_class==2);
HV_aH_class_Z2 = HV_aH_class(HV_aH_class==1);
HV_aH_class_Z3 = HV_aH_class(HV_aH_class==0);
HV_aH_class_Z4 = HV_aH_class(HV_aH_class==5);
HV_aH_class_Z5 = HV_aH_class(HV_aH_class==4);
HV_aH_class_Z6 = HV_aH_class(HV_aH_class==3);
HV_aH_class_Z7 = HV_aH_class(HV_aH_class==8);
HV_aH_class_Z8 = HV_aH_class(HV_aH_class==7);
HV_aH_class_Z9 = HV_aH_class(HV_aH_class==6);

VV_aH_class_Z1 = VV_aH_class(VV_aH_class==2);
VV_aH_class_Z2 = VV_aH_class(VV_aH_class==1);
VV_aH_class_Z3 = VV_aH_class(VV_aH_class==0);
VV_aH_class_Z4 = VV_aH_class(VV_aH_class==5);
VV_aH_class_Z5 = VV_aH_class(VV_aH_class==4);
VV_aH_class_Z6 = VV_aH_class(VV_aH_class==3);
VV_aH_class_Z7 = VV_aH_class(VV_aH_class==8);
VV_aH_class_Z8 = VV_aH_class(VV_aH_class==7);
VV_aH_class_Z9 = VV_aH_class(VV_aH_class==6);

PCO_aH_class_Z1 = PCO_aH_class(PCO_aH_class==2);
PCO_aH_class_Z2 = PCO_aH_class(PCO_aH_class==1);
PCO_aH_class_Z3 = PCO_aH_class(PCO_aH_class==0);
PCO_aH_class_Z4 = PCO_aH_class(PCO_aH_class==5);
PCO_aH_class_Z5 = PCO_aH_class(PCO_aH_class==4);
PCO_aH_class_Z6 = PCO_aH_class(PCO_aH_class==3);
PCO_aH_class_Z7 = PCO_aH_class(PCO_aH_class==8);
PCO_aH_class_Z8 = PCO_aH_class(PCO_aH_class==7);
PCO_aH_class_Z9 = PCO_aH_class(PCO_aH_class==6);

% extracting alpha values
HH_aH_alpha_Z1 = HH_aH_alpha(HH_aH_class==2);
HH_aH_alpha_Z2 = HH_aH_alpha(HH_aH_class==1);
HH_aH_alpha_Z3 = HH_aH_alpha(HH_aH_class==0);
HH_aH_alpha_Z4 = HH_aH_alpha(HH_aH_class==5);
HH_aH_alpha_Z5 = HH_aH_alpha(HH_aH_class==4);
HH_aH_alpha_Z6 = HH_aH_alpha(HH_aH_class==3);
HH_aH_alpha_Z7 = HH_aH_alpha(HH_aH_class==8);
HH_aH_alpha_Z8 = HH_aH_alpha(HH_aH_class==7);
HH_aH_alpha_Z9 = HH_aH_alpha(HH_aH_class==6);

HV_aH_alpha_Z1 = HV_aH_alpha(HV_aH_class==2);
HV_aH_alpha_Z2 = HV_aH_alpha(HV_aH_class==1);
HV_aH_alpha_Z3 = HV_aH_alpha(HV_aH_class==0);
HV_aH_alpha_Z4 = HV_aH_alpha(HV_aH_class==5);
HV_aH_alpha_Z5 = HV_aH_alpha(HV_aH_class==4);
HV_aH_alpha_Z6 = HV_aH_alpha(HV_aH_class==3);
HV_aH_alpha_Z7 = HV_aH_alpha(HV_aH_class==8);
HV_aH_alpha_Z8 = HV_aH_alpha(HV_aH_class==7);
HV_aH_alpha_Z9 = HV_aH_alpha(HV_aH_class==6);

VV_aH_alpha_Z1 = VV_aH_alpha(VV_aH_class==2);
VV_aH_alpha_Z2 = VV_aH_alpha(VV_aH_class==1);
VV_aH_alpha_Z3 = VV_aH_alpha(VV_aH_class==0);
VV_aH_alpha_Z4 = VV_aH_alpha(VV_aH_class==5);
```

```
VV_aH_alpha_Z5 = VV_aH_alpha(VV_aH_class==4);
VV_aH_alpha_Z6 = VV_aH_alpha(VV_aH_class==3);
VV_aH_alpha_Z7 = VV_aH_alpha(VV_aH_class==8);
VV_aH_alpha_Z8 = VV_aH_alpha(VV_aH_class==7);
VV_aH_alpha_Z9 = VV_aH_alpha(VV_aH_class==6);

PCO_aH_alpha_Z1 = PCO_aH_alpha(PCO_aH_class==2);
PCO_aH_alpha_Z2 = PCO_aH_alpha(PCO_aH_class==1);
PCO_aH_alpha_Z3 = PCO_aH_alpha(PCO_aH_class==0);
PCO_aH_alpha_Z4 = PCO_aH_alpha(PCO_aH_class==5);
PCO_aH_alpha_Z5 = PCO_aH_alpha(PCO_aH_class==4);
PCO_aH_alpha_Z6 = PCO_aH_alpha(PCO_aH_class==3);
PCO_aH_alpha_Z7 = PCO_aH_alpha(PCO_aH_class==8);
PCO_aH_alpha_Z8 = PCO_aH_alpha(PCO_aH_class==7);
PCO_aH_alpha_Z9 = PCO_aH_alpha(PCO_aH_class==6);

% extracting entropy values
HH_aH_entropy_Z1 = HH_aH_entropy(HH_aH_class==2);
HH_aH_entropy_Z2 = HH_aH_entropy(HH_aH_class==1);
HH_aH_entropy_Z3 = HH_aH_entropy(HH_aH_class==0);
HH_aH_entropy_Z4 = HH_aH_entropy(HH_aH_class==5);
HH_aH_entropy_Z5 = HH_aH_entropy(HH_aH_class==4);
HH_aH_entropy_Z6 = HH_aH_entropy(HH_aH_class==3);
HH_aH_entropy_Z7 = HH_aH_entropy(HH_aH_class==8);
HH_aH_entropy_Z8 = HH_aH_entropy(HH_aH_class==7);
HH_aH_entropy_Z9 = HH_aH_entropy(HH_aH_class==6);

HV_aH_entropy_Z1 = HV_aH_entropy(HV_aH_class==2);
HV_aH_entropy_Z2 = HV_aH_entropy(HV_aH_class==1);
HV_aH_entropy_Z3 = HV_aH_entropy(HV_aH_class==0);
HV_aH_entropy_Z4 = HV_aH_entropy(HV_aH_class==5);
HV_aH_entropy_Z5 = HV_aH_entropy(HV_aH_class==4);
HV_aH_entropy_Z6 = HV_aH_entropy(HV_aH_class==3);
HV_aH_entropy_Z7 = HV_aH_entropy(HV_aH_class==8);
HV_aH_entropy_Z8 = HV_aH_entropy(HV_aH_class==7);
HV_aH_entropy_Z9 = HV_aH_entropy(HV_aH_class==6);

VV_aH_entropy_Z1 = VV_aH_entropy(VV_aH_class==2);
VV_aH_entropy_Z2 = VV_aH_entropy(VV_aH_class==1);
VV_aH_entropy_Z3 = VV_aH_entropy(VV_aH_class==0);
VV_aH_entropy_Z4 = VV_aH_entropy(VV_aH_class==5);
VV_aH_entropy_Z5 = VV_aH_entropy(VV_aH_class==4);
VV_aH_entropy_Z6 = VV_aH_entropy(VV_aH_class==3);
VV_aH_entropy_Z7 = VV_aH_entropy(VV_aH_class==8);
VV_aH_entropy_Z8 = VV_aH_entropy(VV_aH_class==7);
VV_aH_entropy_Z9 = VV_aH_entropy(VV_aH_class==6);

PCO_aH_entropy_Z1 = PCO_aH_entropy(PCO_aH_class==2);
PCO_aH_entropy_Z2 = PCO_aH_entropy(PCO_aH_class==1);
PCO_aH_entropy_Z3 = PCO_aH_entropy(PCO_aH_class==0);
PCO_aH_entropy_Z4 = PCO_aH_entropy(PCO_aH_class==5);
PCO_aH_entropy_Z5 = PCO_aH_entropy(PCO_aH_class==4);
PCO_aH_entropy_Z6 = PCO_aH_entropy(PCO_aH_class==3);
PCO_aH_entropy_Z7 = PCO_aH_entropy(PCO_aH_class==8);
PCO_aH_entropy_Z8 = PCO_aH_entropy(PCO_aH_class==7);
PCO_aH_entropy_Z9 = PCO_aH_entropy(PCO_aH_class==6);


%% Plot temporal coherence vs. zone for each polarization
```

```
figure;
subplot(3,3,1); hist(HH_pcct(HH_aH_class==8),100);
subplot(3,3,2); hist(HH_pcct(HH_aH_class==5),100);
subplot(3,3,3); hist(HH_pcct(HH_aH_class==2),100);
subplot(3,3,4); hist(HH_pcct(HH_aH_class==7),100);
subplot(3,3,5); hist(HH_pcct(HH_aH_class==4),100);
subplot(3,3,6); hist(HH_pcct(HH_aH_class==1),100);
subplot(3,3,7); hist(HH_pcct(HH_aH_class==6),100);
subplot(3,3,8); hist(HH_pcct(HH_aH_class==3),100);
subplot(3,3,9); hist(HH_pcct(HH_aH_class==0),100);

figure;
subplot(3,3,1); hist(HV_pcct(HV_aH_class==8),100);
subplot(3,3,2); hist(HV_pcct(HV_aH_class==5),100);
subplot(3,3,3); hist(HV_pcct(HV_aH_class==2),100);
subplot(3,3,4); hist(HV_pcct(HV_aH_class==7),100);
subplot(3,3,5); hist(HV_pcct(HV_aH_class==4),100);
subplot(3,3,6); hist(HV_pcct(HV_aH_class==1),100);
subplot(3,3,7); hist(HV_pcct(HV_aH_class==6),100);
subplot(3,3,8); hist(HV_pcct(HV_aH_class==3),100);
subplot(3,3,9); hist(HV_pcct(HV_aH_class==0),100);

figure;
subplot(3,3,1); hist(VV_pcct(VV_aH_class==8),100);
subplot(3,3,2); hist(VV_pcct(VV_aH_class==5),100);
subplot(3,3,3); hist(VV_pcct(VV_aH_class==2),100);
subplot(3,3,4); hist(VV_pcct(VV_aH_class==7),100);
subplot(3,3,5); hist(VV_pcct(VV_aH_class==4),100);
subplot(3,3,6); hist(VV_pcct(VV_aH_class==1),100);
subplot(3,3,7); hist(VV_pcct(VV_aH_class==6),100);
subplot(3,3,8); hist(VV_pcct(VV_aH_class==3),100);
subplot(3,3,9); hist(VV_pcct(VV_aH_class==0),100);

figure;
subplot(3,3,1); hist(PCO_pcct(PCO_aH_class==8),100);
subplot(3,3,2); hist(PCO_pcct(PCO_aH_class==5),100);
subplot(3,3,3); hist(PCO_pcct(PCO_aH_class==2),100);
subplot(3,3,4); hist(PCO_pcct(PCO_aH_class==7),100);
subplot(3,3,5); hist(PCO_pcct(PCO_aH_class==4),100);
subplot(3,3,6); hist(PCO_pcct(PCO_aH_class==1),100);
subplot(3,3,7); hist(PCO_pcct(PCO_aH_class==6),100);
subplot(3,3,8); hist(PCO_pcct(PCO_aH_class==3),100);
subplot(3,3,9); hist(PCO_pcct(PCO_aH_class==0),100);

%% a-H first order stats by zone

disp(['| mean: ' num2str(mean(HH_aH_alpha_Z1)) '| var: ' num2str(var(HH_aH_alpha_Z1)) '| std: '
num2str(std(HH_aH_alpha_Z1))]);
disp(['| mean: ' num2str(mean(HH_aH_alpha_Z2)) '| var: ' num2str(var(HH_aH_alpha_Z2)) '| std: '
num2str(std(HH_aH_alpha_Z2))]);
disp(['| mean: ' num2str(mean(HH_aH_alpha_Z3)) '| var: ' num2str(var(HH_aH_alpha_Z3)) '| std: '
num2str(std(HH_aH_alpha_Z3))]);
disp(['| mean: ' num2str(mean(HH_aH_alpha_Z4)) '| var: ' num2str(var(HH_aH_alpha_Z4)) '| std: '
num2str(std(HH_aH_alpha_Z4))]);
disp(['| mean: ' num2str(mean(HH_aH_alpha_Z5)) '| var: ' num2str(var(HH_aH_alpha_Z5)) '| std: '
num2str(std(HH_aH_alpha_Z5))]);
disp(['| mean: ' num2str(mean(HH_aH_alpha_Z6)) '| var: ' num2str(var(HH_aH_alpha_Z6)) '| std: '
num2str(std(HH_aH_alpha_Z6))]);
```

```
disp(['| mean: ' num2str(mean(HH_aH_alpha_Z7)) '| var: ' num2str(var(HH_aH_alpha_Z7)) '| std: '
num2str(std(HH_aH_alpha_Z7))]);
disp(['| mean: ' num2str(mean(HH_aH_alpha_Z8)) '| var: ' num2str(var(HH_aH_alpha_Z8)) '| std: '
num2str(std(HH_aH_alpha_Z8))]);
disp(['| mean: ' num2str(mean(HH_aH_alpha_Z9)) '| var: ' num2str(var(HH_aH_alpha_Z9)) '| std: '
num2str(std(HH_aH_alpha_Z9))]);

disp(['| mean: ' num2str(mean(HH_aH_entropy_Z1)) '| var: ' num2str(var(HH_aH_entropy_Z1)) '| std: '
num2str(std(HH_aH_entropy_Z1))]);
disp(['| mean: ' num2str(mean(HH_aH_entropy_Z2)) '| var: ' num2str(var(HH_aH_entropy_Z2)) '| std: '
num2str(std(HH_aH_entropy_Z2))]);
disp(['| mean: ' num2str(mean(HH_aH_entropy_Z3)) '| var: ' num2str(var(HH_aH_entropy_Z3)) '| std: '
num2str(std(HH_aH_entropy_Z3))]);
disp(['| mean: ' num2str(mean(HH_aH_entropy_Z4)) '| var: ' num2str(var(HH_aH_entropy_Z4)) '| std: '
num2str(std(HH_aH_entropy_Z4))]);
disp(['| mean: ' num2str(mean(HH_aH_entropy_Z5)) '| var: ' num2str(var(HH_aH_entropy_Z5)) '| std: '
num2str(std(HH_aH_entropy_Z5))]);
disp(['| mean: ' num2str(mean(HH_aH_entropy_Z6)) '| var: ' num2str(var(HH_aH_entropy_Z6)) '| std: '
num2str(std(HH_aH_entropy_Z6))]);
disp(['| mean: ' num2str(mean(HH_aH_entropy_Z7)) '| var: ' num2str(var(HH_aH_entropy_Z7)) '| std: '
num2str(std(HH_aH_entropy_Z7))]);
disp(['| mean: ' num2str(mean(HH_aH_entropy_Z8)) '| var: ' num2str(var(HH_aH_entropy_Z8)) '| std: '
num2str(std(HH_aH_entropy_Z8))]);
disp(['| mean: ' num2str(mean(HH_aH_entropy_Z9)) '| var: ' num2str(var(HH_aH_entropy_Z9)) '| std: '
num2str(std(HH_aH_entropy_Z9))]);

disp(['| mean: ' num2str(mean(HV_aH_alpha_Z1)) '| var: ' num2str(var(HV_aH_alpha_Z1)) '| std: '
num2str(std(HV_aH_alpha_Z1))]);
disp(['| mean: ' num2str(mean(HV_aH_alpha_Z2)) '| var: ' num2str(var(HV_aH_alpha_Z2)) '| std: '
num2str(std(HV_aH_alpha_Z2))]);
disp(['| mean: ' num2str(mean(HV_aH_alpha_Z3)) '| var: ' num2str(var(HV_aH_alpha_Z3)) '| std: '
num2str(std(HV_aH_alpha_Z3))]);
disp(['| mean: ' num2str(mean(HV_aH_alpha_Z4)) '| var: ' num2str(var(HV_aH_alpha_Z4)) '| std: '
num2str(std(HV_aH_alpha_Z4))]);
disp(['| mean: ' num2str(mean(HV_aH_alpha_Z5)) '| var: ' num2str(var(HV_aH_alpha_Z5)) '| std: '
num2str(std(HV_aH_alpha_Z5))]);
disp(['| mean: ' num2str(mean(HV_aH_alpha_Z6)) '| var: ' num2str(var(HV_aH_alpha_Z6)) '| std: '
num2str(std(HV_aH_alpha_Z6))]);
disp(['| mean: ' num2str(mean(HV_aH_alpha_Z7)) '| var: ' num2str(var(HV_aH_alpha_Z7)) '| std: '
num2str(std(HV_aH_alpha_Z7))]);
disp(['| mean: ' num2str(mean(HV_aH_alpha_Z8)) '| var: ' num2str(var(HV_aH_alpha_Z8)) '| std: '
num2str(std(HV_aH_alpha_Z8))]);
disp(['| mean: ' num2str(mean(HV_aH_alpha_Z9)) '| var: ' num2str(var(HV_aH_alpha_Z9)) '| std: '
num2str(std(HV_aH_alpha_Z9))]);

disp(['| mean: ' num2str(mean(HV_aH_entropy_Z1)) '| var: ' num2str(var(HV_aH_entropy_Z1)) '| std: '
num2str(std(HV_aH_entropy_Z1))]);
disp(['| mean: ' num2str(mean(HV_aH_entropy_Z2)) '| var: ' num2str(var(HV_aH_entropy_Z2)) '| std: '
num2str(std(HV_aH_entropy_Z2))]);
disp(['| mean: ' num2str(mean(HV_aH_entropy_Z3)) '| var: ' num2str(var(HV_aH_entropy_Z3)) '| std: '
num2str(std(HV_aH_entropy_Z3))]);
disp(['| mean: ' num2str(mean(HV_aH_entropy_Z4)) '| var: ' num2str(var(HV_aH_entropy_Z4)) '| std: '
num2str(std(HV_aH_entropy_Z4))]);
disp(['| mean: ' num2str(mean(HV_aH_entropy_Z5)) '| var: ' num2str(var(HV_aH_entropy_Z5)) '| std: '
num2str(std(HV_aH_entropy_Z5))]);
disp(['| mean: ' num2str(mean(HV_aH_entropy_Z6)) '| var: ' num2str(var(HV_aH_entropy_Z6)) '| std: '
num2str(std(HV_aH_entropy_Z6))]);
disp(['| mean: ' num2str(mean(HV_aH_entropy_Z7)) '| var: ' num2str(var(HV_aH_entropy_Z7)) '| std: '
num2str(std(HV_aH_entropy_Z7))]);
```

```
disp(['| mean: ' num2str(mean(HV_aH_entropy_Z8)) '| var: ' num2str(var(HV_aH_entropy_Z8)) '| std: '
num2str(std(HV_aH_entropy_Z8))]);
disp(['| mean: ' num2str(mean(HV_aH_entropy_Z9)) '| var: ' num2str(var(HV_aH_entropy_Z9)) '| std: '
num2str(std(HV_aH_entropy_Z9))]);

disp(['| mean: ' num2str(mean(VV_aH_alpha_Z1)) '| var: ' num2str(var(VV_aH_alpha_Z1)) '| std: '
num2str(std(VV_aH_alpha_Z1))]);
disp(['| mean: ' num2str(mean(VV_aH_alpha_Z2)) '| var: ' num2str(var(VV_aH_alpha_Z2)) '| std: '
num2str(std(VV_aH_alpha_Z2))]);
disp(['| mean: ' num2str(mean(VV_aH_alpha_Z3)) '| var: ' num2str(var(VV_aH_alpha_Z3)) '| std: '
num2str(std(VV_aH_alpha_Z3))]);
disp(['| mean: ' num2str(mean(VV_aH_alpha_Z4)) '| var: ' num2str(var(VV_aH_alpha_Z4)) '| std: '
num2str(std(VV_aH_alpha_Z4))]);
disp(['| mean: ' num2str(mean(VV_aH_alpha_Z5)) '| var: ' num2str(var(VV_aH_alpha_Z5)) '| std: '
num2str(std(VV_aH_alpha_Z5))]);
disp(['| mean: ' num2str(mean(VV_aH_alpha_Z6)) '| var: ' num2str(var(VV_aH_alpha_Z6)) '| std: '
num2str(std(VV_aH_alpha_Z6))]);
disp(['| mean: ' num2str(mean(VV_aH_alpha_Z7)) '| var: ' num2str(var(VV_aH_alpha_Z7)) '| std: '
num2str(std(VV_aH_alpha_Z7))]);
disp(['| mean: ' num2str(mean(VV_aH_alpha_Z8)) '| var: ' num2str(var(VV_aH_alpha_Z8)) '| std: '
num2str(std(VV_aH_alpha_Z8))]);
disp(['| mean: ' num2str(mean(VV_aH_alpha_Z9)) '| var: ' num2str(var(VV_aH_alpha_Z9)) '| std: '
num2str(std(VV_aH_alpha_Z9))]);

disp(['| mean: ' num2str(mean(VV_aH_entropy_Z1)) '| var: ' num2str(var(VV_aH_entropy_Z1)) '| std: '
num2str(std(VV_aH_entropy_Z1))]);
disp(['| mean: ' num2str(mean(VV_aH_entropy_Z2)) '| var: ' num2str(var(VV_aH_entropy_Z2)) '| std: '
num2str(std(VV_aH_entropy_Z2))]);
disp(['| mean: ' num2str(mean(VV_aH_entropy_Z3)) '| var: ' num2str(var(VV_aH_entropy_Z3)) '| std: '
num2str(std(VV_aH_entropy_Z3))]);
disp(['| mean: ' num2str(mean(VV_aH_entropy_Z4)) '| var: ' num2str(var(VV_aH_entropy_Z4)) '| std: '
num2str(std(VV_aH_entropy_Z4))]);
disp(['| mean: ' num2str(mean(VV_aH_entropy_Z5)) '| var: ' num2str(var(VV_aH_entropy_Z5)) '| std: '
num2str(std(VV_aH_entropy_Z5))]);
disp(['| mean: ' num2str(mean(VV_aH_entropy_Z6)) '| var: ' num2str(var(VV_aH_entropy_Z6)) '| std: '
num2str(std(VV_aH_entropy_Z6))]);
disp(['| mean: ' num2str(mean(VV_aH_entropy_Z7)) '| var: ' num2str(var(VV_aH_entropy_Z7)) '| std: '
num2str(std(VV_aH_entropy_Z7))]);
disp(['| mean: ' num2str(mean(VV_aH_entropy_Z8)) '| var: ' num2str(var(VV_aH_entropy_Z8)) '| std: '
num2str(std(VV_aH_entropy_Z8))]);
disp(['| mean: ' num2str(mean(VV_aH_entropy_Z9)) '| var: ' num2str(var(VV_aH_entropy_Z9)) '| std: '
num2str(std(VV_aH_entropy_Z9))]);

disp(['| mean: ' num2str(mean(PCO_aH_alpha_Z1)) '| var: ' num2str(var(PCO_aH_alpha_Z1)) '| std: '
num2str(std(PCO_aH_alpha_Z1))]);
disp(['| mean: ' num2str(mean(PCO_aH_alpha_Z2)) '| var: ' num2str(var(PCO_aH_alpha_Z2)) '| std: '
num2str(std(PCO_aH_alpha_Z2))]);
disp(['| mean: ' num2str(mean(PCO_aH_alpha_Z3)) '| var: ' num2str(var(PCO_aH_alpha_Z3)) '| std: '
num2str(std(PCO_aH_alpha_Z3))]);
disp(['| mean: ' num2str(mean(PCO_aH_alpha_Z4)) '| var: ' num2str(var(PCO_aH_alpha_Z4)) '| std: '
num2str(std(PCO_aH_alpha_Z4))]);
disp(['| mean: ' num2str(mean(PCO_aH_alpha_Z5)) '| var: ' num2str(var(PCO_aH_alpha_Z5)) '| std: '
num2str(std(PCO_aH_alpha_Z5))]);
disp(['| mean: ' num2str(mean(PCO_aH_alpha_Z6)) '| var: ' num2str(var(PCO_aH_alpha_Z6)) '| std: '
num2str(std(PCO_aH_alpha_Z6))]);
disp(['| mean: ' num2str(mean(PCO_aH_alpha_Z7)) '| var: ' num2str(var(PCO_aH_alpha_Z7)) '| std: '
num2str(std(PCO_aH_alpha_Z7))]);
disp(['| mean: ' num2str(mean(PCO_aH_alpha_Z8)) '| var: ' num2str(var(PCO_aH_alpha_Z8)) '| std: '
num2str(std(PCO_aH_alpha_Z8))]);
```

```
disp(['| mean: ' num2str(mean(PCO_aH_alpha_Z9)) '| var: ' num2str(var(PCO_aH_alpha_Z9)) '| std: '
num2str(std(PCO_aH_alpha_Z9))]);

disp(['| mean: ' num2str(mean(PCO_aH_entropy_Z1)) '| var: ' num2str(var(PCO_aH_entropy_Z1)) '| std: '
num2str(std(PCO_aH_entropy_Z1))]);
disp(['| mean: ' num2str(mean(PCO_aH_entropy_Z2)) '| var: ' num2str(var(PCO_aH_entropy_Z2)) '| std: '
num2str(std(PCO_aH_entropy_Z2))]);
disp(['| mean: ' num2str(mean(PCO_aH_entropy_Z3)) '| var: ' num2str(var(PCO_aH_entropy_Z3)) '| std: '
num2str(std(PCO_aH_entropy_Z3))]);
disp(['| mean: ' num2str(mean(PCO_aH_entropy_Z4)) '| var: ' num2str(var(PCO_aH_entropy_Z4)) '| std: '
num2str(std(PCO_aH_entropy_Z4))]);
disp(['| mean: ' num2str(mean(PCO_aH_entropy_Z5)) '| var: ' num2str(var(PCO_aH_entropy_Z5)) '| std: '
num2str(std(PCO_aH_entropy_Z5))]);
disp(['| mean: ' num2str(mean(PCO_aH_entropy_Z6)) '| var: ' num2str(var(PCO_aH_entropy_Z6)) '| std: '
num2str(std(PCO_aH_entropy_Z6))]);
disp(['| mean: ' num2str(mean(PCO_aH_entropy_Z7)) '| var: ' num2str(var(PCO_aH_entropy_Z7)) '| std: '
num2str(std(PCO_aH_entropy_Z7))]);
disp(['| mean: ' num2str(mean(PCO_aH_entropy_Z8)) '| var: ' num2str(var(PCO_aH_entropy_Z8)) '| std: '
num2str(std(PCO_aH_entropy_Z8))]);
disp(['| mean: ' num2str(mean(PCO_aH_entropy_Z9)) '| var: ' num2str(var(PCO_aH_entropy_Z9)) '| std: '
num2str(std(PCO_aH_entropy_Z9))]);
```

read_gamma_slc.m

```
function [X, dims] = read_gamma_slc( img_name, par_name )

dims = {[] []};

fid = fopen( par_name );
if fid == -1, error('Could not open file.'), end

% scan parameter file for image dimensions
```

```
while 1
   textline = fgetl( fid );
   if (textline == -1)
      break;
   elseif (~isempty(textline))
      if (isempty(dims{1}))
         dims(1) = textscan( textline, 'azimuth_lines: %d');
      end
      if (isempty(dims{2}))
         dims(2) = textscan( textline, 'range_samples: %d');
      end
   end
end
fclose(fid);

dims = cell2mat(dims);

% read image data from file
X = zeros( dims, 'single' );

fid = fopen( img_name,'r','b' );
if fid == -1, error('Could not open file.'), end

for r = 1:size(X,1)
   A = fread( fid, [2,size(X,2)], 'int16' );
   X(r,:) = A(1,:) + j*A(2,:);
end
fclose(fid);
```

read_gamma_data.m

```
% Read Gamma image data
function X = read_gamma_data( filename, dims )

% read image data from file
[null1,null2, ext] = fileparts(filename);
```

```
% determine datatype from file extension
if strcmpi(ext, '.int') || strcmpi(ext, '.flt') || strcmpi(ext, '.sm') || strcmpi(ext, '.slope')

   X = zeros( dims, 'single' );

   fid = fopen( filename ,'r','b' );
   for r = 1:size(X,1)
      A = fread( fid, [2,size(X,2)], 'int16' );
      X(r,:) = A(1,:) + j*A(2,:);
   end
   fclose(fid);

else  % .cc, .msr, .pwr, .hgt, ...

   X = zeros( dims, 'single' );

   fid = fopen( filename ,'r','b' );
   for r = 1:size(X,1)
      X(r,:) = fread( fid, [1,size(X,2)], 'float32' );
   end
   fclose(fid);

end
```

read_gamma_plist.m

```
function A = read_gamma_plist( file_name )

% read image data from file,
% resulting points coords are in (x,y), or (col,row), format

fid = fopen( file_name,'r','b' );
if fid == -1, error('Could not open file.'), end

A = fread( fid, inf, 'int32' );
A = single(reshape(A,2,[]));
A = A.';

fclose(fid);
```

read_envi_data.m

```
% Read ENVI image data
function X = read_envi_data( filename, envi_datatype, dims )

% ### IMPORTANT NOTE ###
```

```
% data must be MSBF (datatype = 1)
%  so... SARscape generated files need to be converted from LSBF to MSBF

% determine datatype from file extension

% 1=8 bit byte; 2=16-bit signed integer; 3=32-bit signed long integer;
% 4=32-bit floating point; 5=64- bit double precision floating point;
% 6=2x32-bit complex, real-imaginary pair of double precision;
% 9=2x64-bit double precision complex, real-imaginary pair of double precision;
% 12=16-bit unsigned integer; 13=32-bit unsigned long integer;
% 14=64-bit unsigned integer; and 15=64-bit unsigned long integer

switch( envi_datatype )

   case 1

      X = zeros( dims, 'int8' );

      fid = fopen( filename ,'r','b' );
      for r = 1:size(X,1)
         X(r,:) = fread( fid, [1,size(X,2)], 'int8' );

      end
      fclose(fid);

   case 2

      X = zeros( dims, 'int16' );

      fid = fopen( filename ,'r','b' );
      for r = 1:size(X,1)
         X(r,:) = fread( fid, [1,size(X,2)], 'int16' );

      end
      fclose(fid);

   case 3

      X = zeros( dims, 'int32' );

      fid = fopen( filename ,'r','b' );
      for r = 1:size(X,1)
         X(r,:) = fread( fid, [1,size(X,2)], 'int32' );

      end
      fclose(fid);

   case 4

      X = zeros( dims, 'single' );

      fid = fopen( filename ,'r','b' );
      for r = 1:size(X,1)
         X(r,:) = fread( fid, [1,size(X,2)], 'single' );

      end
      fclose(fid);

   case 5
```

```
        X = zeros( dims, 'double' );

        fid = fopen( filename ,'r','b' );
        for r = 1:size(X,1)
           X(r,:) = fread( fid, [1,size(X,2)], 'double' );

        end
        fclose(fid);

    case 6

        X = zeros( dims, 'double' );

        fid = fopen( filename ,'r','b' );
        for r = 1:size(X,1)
           A = fread( fid, [2,size(X,2)], 'float32' );
           X(r,:) = A(1,:) + j*A(2,:);
        end
        fclose(fid);

end
```