

Institut für Geodäsie und Geoinformation

Professur für Theoretische Geodäsie

Entwicklung von Datenstrukturen zur Darstellung
der raum/zeitlichen Zusammenhänge zwischen
kohärenten Pixeln in D-InSAR-Bildstapeln

Masterarbeit

im Masterstudiengang Geodäsie und Geoinformation
an der Hohen Landwirtschaftlichen Fakultät
der Rheinischen Friedrich–Wilhelms–Universität
zu Bonn

vorgelegt am 10. Dezember 2015 von

Joël Noah Köhler

aus Troisdorf

Bonn 2015



Erster Prüfer: Prof. Dr. techn. W.-D. Schuh

Zweiter Prüfer: M.Sc. C. Esch

Masterarbeit Joël Noah Köhler

28. April 2015

Thema: Entwicklung von Datenstrukturen zur Darstellung der raum/zeitlichen Zusammenhänge zwischen kohärenten Pixeln in dInSAR Bildstapeln

Die differentielle Auswertung von interferometrischen Satelliten-Radardaten (dInSAR) eröffnet die Möglichkeit großräumige und kleinräumige Deformationen der Erdoberfläche zu beobachten. Während neue Satellitenmissionen wie Sentinel-1 eine sehr gute räumliche und zeitliche Überdeckung bieten, kommt es bei der Auswertung von älteren Satellitenmissionen (ERS, Envisat) speziell in ländlichen Gebieten zu erheblichen Problemen, da die Aufnahmen nur wenige kohärente Gebiete aufweisen. Durch die Auswertung dieser älteren Messungen können aber Zeitreihen mit bis zu 25 Jahren aufgebaut werden und damit zusätzliche Erkenntnisse abgeleitet werden.

Ein maßgebender Prozessierungsschritt für die Genauigkeit und Richtigkeit dieser Datenprodukte, ist das Lösen der Phasenmehrdeutigkeiten. Das Verfahren der Phasenabwicklung (Phase Unwrapping) für eine dInSAR-Szene ist ein schlecht gestelltes Problem und ohne zusätzliche Annahmen nicht eindeutig lösbar (Chen und Zebker, 2001). Durch das Stapeln von differentiellen radarinterferometrischen Messungen (dInSAR) können zusätzliche Information über das zeitliche Verhalten bei der Phasenabwicklung genutzt werden. In den letzten Jahren wurden eine Vielzahl von Algorithmen zur Lösung dieses Problems entwickelt, welche sich grundsätzlich in zwei- und dreidimensionale Verfahren unterteilen. Hierbei ist das Ziel dieser Verfahren die Bestimmung eines zusammenhängenden Phasenfeldes, woraus dann durch einen geeigneten Algorithmus (z.B. das SBAS Verfahren von Lanari et.al. (2007)) Deformationen der Erdoberfläche ableitbar sind.

In dieser Masterarbeit geht es zunächst um den Aufbau einer geeigneten Graphenstruktur zur Darstellung der raum/zeitlichen Zusammenhänge zwischen kohärenten Pixeln in dInSAR Bildstapeln. Typische Anwendungen, wie die Abfrage von Knoten/Kanten/Flächen/Volumen-Strukturen in zeitlicher, räumlicher und raum/zeitlicher Form sollen effizient möglich sein. Für die Verknüpfung der Daten (Kantenerstellung) sollen unterschiedliche Algorithmen implementiert werden, die sowohl die Geometrie als auch zusätzliche Attribute wie die zeitliche und räumliche Basislinie berücksichtigen können. Neben der strengen Dreiecksvermaschung (z.B. durch Delaunay Triangulation) sollen zusätzliche Optionen möglich sein. So soll die Einführung zusätzlicher Kanten (z.B. für kurze Basislinien) bzw. die Elimination von Kanten (bei langen Basislinien) individuell erfolgen können. Das Programmpaket soll in der Lage sein auch mit diesen entarteten Flächen (überlappende Dreiecke, bzw. Vielecke) und Situationen umgehen zu können. Abfragen bezüglich Knoten, Kanten und Flächen sollen sowohl als Einzelabfragen als auch als Gesamtliste erfolgen können. Die Ausgaben sollen dabei alphanumerisch erfolgen und durch entsprechende graphische Darstellungen unterstützt werden.

Ziel der Arbeit ist die Erarbeitung der Datenstrukturen und der entsprechenden Abfragen und deren ausführliche Dokumentation. Die Funktionsfähigkeit der Implementierung soll am dInSAR-ERS-Datenstapel der Niederrheinischen Bucht demonstriert werden.

Literatur

Chen, C. W. und Zebker, H. A. (2001) *Two-dimensional phase unwrapping with use of statistical models for cost functions in nonlinear optimization*. JOSA A 18(2): S. 338-351, Optical Society of America.

Lanari, R.; Casu, F.; Manzo, M.; Zeni, G.; Berardino, P.; Manunta, M. und Pepe, A. An Overview of the Small Baseline Subset Algorithm: A DInSAR Technique for Surface Deformation Analysis Pure and Applied Geophysics, Birkhäuser-Verlag, 2007, 164, 637-661

Aufgabensteller: Prof. Dr. techn. Wolf-Dieter Schuh
Betreuer: Christina Esch MSc.

Tag der Ausgabe:
Dauer: 6 Monate
Tag der Abgabe:

Bonn, den 28. April 2015


Wolf-Dieter Schuh

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht.

Bonn, 10. Dezember 2015

Joël Noah Köhler

Zusammenfassung

Durch verschiedene Ursachen kommt es auf der ganzen Welt zu Deformationen der Erdoberfläche. In der Region der Niederrheinischen Bucht sind die Senkungen der Landesfläche von NRW beispielsweise auf die Braunkohlegewinnung zurückzuführen. Grund dafür ist der Sumpfungsprozess, bei dem das Grundwasser zur Trockenlegung der Tagebaue bis zum Abbauniveau abgepumpt wird. Zur Überwachung dieser Oberflächenveränderungen werden in einem Vier-Jahre-Zyklus Leitnivellements durchgeführt. Eine weitere Möglichkeit zur Detektion der Deformationen bietet die differentielle Radarinterferometrie (D-InSAR). Dazu wird ein interferometrisches Phasenbild aus zwei SAR-Szenen der gleichen Region erstellt, die zu unterschiedlichen Zeitpunkten aufgenommen wurden. Von diesem Bild wird anschließend der Einfluss der Topographie unter Hinzunahme eines Digitalen Geländemodells abgezogen. Bei geeigneter Auswertung kann aus solchen D-InSAR-Aufnahmen die Deformation der Erdoberfläche zwischen den beiden Aufnahmezeitpunkten bestimmt werden.

Als Datengrundlage liegt dieser Arbeit ein Bildstapel mit 191 differentiellen Interferogrammen vor, die aus 71 SAR-Aufnahmen der *European Remote sensing Satellite* Missionen (ERS-1/2) erzeugt wurden. Daher wird das Konzept der differentiellen Radarinterferometrie in dieser Arbeit kurz erläutert. Ein besonderer Augenmerk wird dabei auf die Kohärenz gerichtet, die ein genauigkeitsbestimmendes Merkmal für die Phasenbilder darstellt. Bei abnehmender Kohärenz nimmt das Phasenrauschen in den Interferogrammen zu, was die Auswertbarkeit von Pixeln beschränkt. Es wird gezeigt, dass die Abnahme der Kohärenz auf verschiedene Effekte zurückgeführt werden kann, die unter anderem von der Auswahl der Datenpaare zur Erzeugung von D-InSAR-Bildern abhängen. Des Weiteren werden unterschiedliche Möglichkeiten zum Aufbau eines D-InSAR-Bildstapels vorgestellt. Diese werden durch verschiedene Methoden zur Deformationsbestimmung motiviert.

Ein maßgeblicher Prozessierungsschritt, der die Genauigkeit und Richtigkeit der Auswertung beeinflusst, ist das Lösen der Phasenmehrdeutigkeiten (*engl.* Phase Unwrapping). Daher werden die Grundlagen dieses Bearbeitungsschrittes in dieser Arbeit zusammengefasst. Weiterhin wird auf zwei spezielle Phase Unwrapping Methoden eingegangen, die für das Lösen von Phasenmehrdeutigkeiten in Bildstapeln entwickelt wurden. Der EMCF-Algorithmus von PEPE UND LANARI (2006) nutzt dabei die räumlichen und zeitlichen Zusammenhänge zwischen den Interferogrammen im Bildstapel, um das Phase Unwrapping robuster gegenüber Fehlern zu machen. Hingegen setzt der Edgelist-Algorithmus auf eine direkte Verknüpfung von Punkten zwischen benachbarten Bildern im Stapel, um die Robustheit zu steigern (SHANKER UND ZEBKER, 2010). Mit der SBAS-Technik wird ein stapelbasiertes Verfahren vorgestellt, das nach der Lösung der Phasenmehrdeutigkeiten die Deformationen der Erdoberfläche bestimmen kann (BERARDINO ET AL., 2002).

Aus den Eigenschaften der Bilder und des Bildstapels sowie den präsentierten Anwendungen werden Anforderungen an eine Datenstruktur abgeleitet. Mit dieser soll es möglich sein, die komplexen räumlichen und zeitlichen Zusammenhänge von kohärenten Pixeln in D-InSAR-Bildstapeln darzustellen. Eine solche Struktur wurde im Laufe dieser Arbeit entwickelt, wobei hierfür ein objektorientierter Ansatz gewählt wurde. Die Datenstruktur besteht aus zwei Substrukturen, die als *Zeitliche* bzw. *Räumlichen Struktur* bezeichnet werden. In der ersten der beiden werden die Zusammenhänge zwischen den Interferogrammen eines Bildstapels dargestellt. Im Gegensatz dazu verknüpft die Räumliche Struktur die kohärenten Pixel miteinander. Verwaltet wird die Struktur von einer Klasse, die den gesamten Bildstapel repräsentiert. Sie dient gleichzeitig als Nutzerschnittstelle, mit der auch übergreifende Anfragen an die Substrukturen gestellt werden können. Anhand von UML-Diagrammen wird das hier entwickelte Modell dieser Datenstruktur ausführlich dokumentiert.

Abschließend wird noch die Lauffähigkeit der Implementierung dieser Datenstruktur nachgewiesen. Dazu wird aus der Datengrundlage ein Testdatensatz abgeleitet. Mit diesem werden die umgesetzten Anfragen an die Datenstruktur ausgeführt und in einem Funktionshandbuch veranschaulicht.

Abstract

The deformation of the earth's surface is a global phenomenon which refers to various causes. For example, in the region of the Niederrheinische Bucht, the surface of North Rhine Westphalia is subsiding. This is due to the brown coal mining in this area. To be more precise, it is caused by the so called sump dewatering, which means the drawdown of the groundwater level for the drainage of the opencast mines. In order to monitor these changes in altitude, leveling campaigns are performed in a cycle of four years. Another option for the detection of these deformations is based on the differential radar interferometry (D-InSAR). For this purpose, an interferometric phase image is generated from two SAR images of the same area, which are acquired at two different points in time. Subsequently, the influence of the topography is removed from the phases with the help of a digital elevation model. With an appropriate evaluation of a D-InSAR image, it is possible to extract the surface changes between the two acquisition dates.

The data basis of this thesis is an image stack consisting of 191 differential interferograms. These are generated from 71 SAR acquisitions of the *European Remote sensing Satellite* missions (ERS-1/2). Hence, the basics of the differential radar interferometry are summarized in this thesis, with a specific focus on the coherence. This factor defines the quality of a phase image, since the phase noise increases with decreasing coherence. Due to this, the usability of some pixels is limited or even non-existing. It is shown that the decrease has several causes, which are partially related to the choice of the SAR acquisition that form a D-InSAR image. Furthermore, different options are presented to build an image stack. These options are motivated by various methods for the computation of surface deformations from the corresponding stacks. Solving the phase ambiguities is an essential step in the processing of D-InSAR images, as the accuracy and correctness of the final results depend on it. Generally, this step is called phase unwrapping. That is why the basics of this operation are outlined in this thesis. Additionally, we will elaborate on two specific phase unwrapping methods, which are developed especially for solving the phase ambiguities in image stacks. The usage of the time component of a D-InSAR stack makes the phase unwrapping more robust against processing errors. For this purpose, the so called EMCF algorithm by PEPE UND LANARI (2006) exploits the dependencies amongst the interferograms. However, the edgelist algorithm by SHANKER UND ZEBKER (2010) links coherent pixels, with the same coordinates, in adjacent images of the stack. With the SBAS technique, a stack-based method will be introduced that applies the unwrapped interferograms to compute the surface deformations (BERARDINO ET AL., 2002).

The outlined characteristics of differential interferograms and the image stack together with the depicted applications will be used to impose requirements on a data structure. This structure is supposed to describe the temporal and spatial dependencies of coherent pixels in an image stack consisting of differential interferograms. Such a structure was developed in course of this thesis using an object-oriented approach. The data structure consists of two substructures, which are referred to as *Temporal* and *Spatial Structure*. The first of them models the dependencies amongst the interferograms whereas the second links the coherent pixels in the image stack. An additional class, representing the image stack as such, is created to govern the two substructures. Simultaneously, this class is the user interface, which allows to raise requests even across the substructures. The developed model is documented exhaustively with the help of UML diagrams.

Finally, the executability of the implementation of this data structure is demonstrated. To this end, a test case is derived from the data basis. Subsequently, this test case is used to run the functions of the package. The results are visualized in the form of a function manual.

Inhaltsverzeichnis

1	Einleitung	3
1.1	Motivation	3
1.2	Stand der Forschung	4
1.3	Wissenschaftliche Zielsetzung der Arbeit	5
1.4	Gliederung der Arbeit	6
2	Grundlagen für die Beschreibung von D-InSAR-Bildstapeln	7
2.1	Radar mit synthetischer Apertur	7
2.2	Radarinterferometrie	8
2.3	Differentielle Radarinterferometrie	9
2.4	Kohärenz	9
2.5	Erzeugung eines Bildstapels von differentiellen SAR-Interferogrammen	13
2.5.1	Methoden zur Selektion von Datenpaaren	14
2.5.2	Beschreibung der Datengrundlage	17
3	Prozessierung von D-InSAR-Bildstapel	21
3.1	Allgemeine Grundlagen des Phase Unwrappings	21
3.2	Darstellung des Phase Unwrappings als MCF-Problem	22
3.3	Der Extended Minimum Cost Flow Algorithmus	24
3.3.1	Temporal Unwrapping	25
3.3.2	Spatial Unwrapping	27
3.4	Der Edgelist Algorithmus	27
3.5	Die Small BAseline Subset-Technik	29
4	Aufbau und Funktionalitäten der Datenstrukturen	32
4.1	Anforderungen an die Datenstrukturen	32
4.2	Modellierung der Datenstrukturen	33
4.2.1	Modell des Bildstapels	33
4.2.2	Modell der Zeitlichen Struktur	34
4.2.3	Modell der Räumlichen Struktur	41
4.3	Anfragen an die Datenstrukturen	48

5 Implementierung	51
5.1 Auswahl eines Tesdatensatzes	51
5.2 Funktionsübersicht	55
6 Schlussbemerkungen	56
6.1 Fazit	56
6.2 Ausblick	57
7 Danksagung	58
A Bezeichnungen	i
B Abkürzungen	iv
C Funktionshandbuch	v
Verzeichnisse	xxiv
Abbildungsverzeichnis	xxiv
Tabellenverzeichnis	xxvi
Literatur	xxvii

1. Einleitung

1.1 Motivation

Die differentielle Radarinterferometrie (D-InSAR) basiert auf der Auswertung der Phasendifferenz zweier Satelliten-Radaraufnahmen, von der zusätzlich der Einfluss eines Höhenmodells abgezogen wird. Mit den so entstehenden D-InSAR-Bildern ist es möglich, großräumige und kleinräumige Deformationen der Erdoberfläche im Millimeter- bis Zentimeterbereich zu detektieren.

Gerade in der Niederrheinischen Bucht, die von den Städten Aachen, Euskirchen, Köln und Mönchengladbach begrenzt wird, kommt es durch den Braunkohleabbau zu Bodensenkungen. Die daraus resultierenden Deformationen sind auf die Kohlegewinnung in den Tagebauen Garzweiler, Hambach und Inden zurückzuführen, da hier das Grundwasser bis unter die Abbautiefe abgepumpt werden muss.

Durch Leitnivelements in der Region werden diese Deformationen in einem Vier-Jahre-Zyklus bestimmt. Die zeitliche Abtastung durch Radar-Satelliten ist jedoch deutlich höher. Daher ist es von großem Interesse, eine Deformationszeitreihe aus D-InSAR-Daten aufzubauen.

In Abbildung 1.1 ist ein Überblick über vergangene, aktuelle und künftige europäische SAR-Missionen dargestellt. Wie der Grafik zu entnehmen ist, lässt sich mit den Radardaten dieser Missionen eine Zeitreihe von ca. 25 Jahren aufbauen. Die neueren Satelliten (TerraSAR-X, TanDEM-X und Sentinel-1A) bieten eine hohe räumliche und zeitliche Abdeckung. Jedoch bilden gerade die älteren Missionen (ERS-1/2 und Envisat) die längste Zeitspanne für eine solche Zeitreihe.

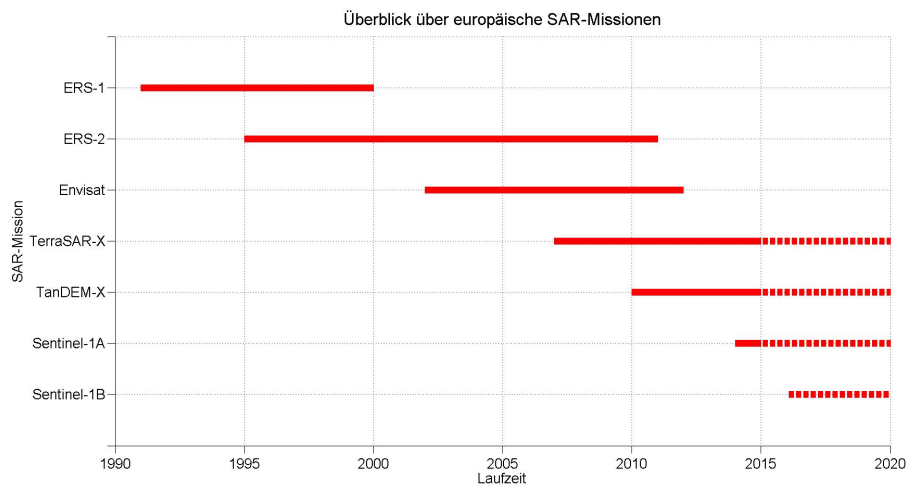


Abbildung 1.1: Überblick über europäische SAR-Missionen.

Die Auswertung der D-InSAR-Bilder dieser Missionen kann sich als problematisch herausstellen. Denn in Gebieten, die von agrar- oder forstwirtschaftlichen Flächen geprägt sind, kommt es zu einer starken Abnahme der Kohärenz. Im Interferogramm macht sich dies durch ein starkes Phasenrauschen bemerkbar, sodass dort eine Auswertung nicht möglich ist. Dies gilt auch im Speziellen für Interferogramme der Niederrheinischen Bucht, wie an Abbildung 1.2 zu erkennen ist. Es wird deutlich, dass ein Großteil des Phasenbildes von Rauschen gekennzeichnet ist.

Ein genauigkeitsbestimmender Prozessierungsschritt zur Bestimmung der Deformationen ist das Lösen der Phasenmehrdeutigkeiten. Dies stellt ein nicht lösbares Problem dar, das nur durch das Treffen zusätzlicher Annahmen bewältigt werden kann (CHEN, 2001, S. 8). In den vergangenen Jahren wurden spezielle Techni-

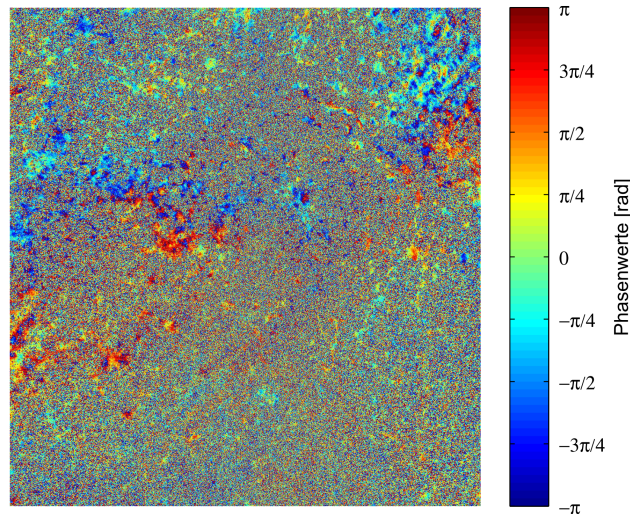


Abbildung 1.2: Differenzielles Phasenbild abgeleitet aus den SAR-Aufnahmen der ERS-Satelliten zu den Aufnahmezeitpunkten 18.07.1992 und 29.05.1993.

ken für die sogenannte Phasenabwicklung (*engl.* Phase Unwrapping) entwickelt, die mit ganzen Bildstapeln arbeiten. Dadurch sollen zusätzliche Informationen nutzbar werden, die sich aus den Zusammenhängen zwischen den D-InSAR-Bildern ableiten lassen. Die Hinzunahme dieser Informationen beim Phase Unwrapping soll die Verfahren robuster machen. Zu diesen Applikationen gehören beispielsweise der EMCF-Algorithmus von PEPE UND LANARI (2006) und der Edgelist-Algorithmus nach SHANKER UND ZEBKER (2010). Nach dem Lösen der Phasenmehrdeutigkeiten stellt die SBAS-Methode von BERARDINO ET AL. (2002) eine mögliche Anwendung dar, die aus einem Bildstapel von abgewickelten D-InSAR-Aufnahmen die gesuchten Deformationen bestimmt.

1.2 Stand der Forschung

Im Laufe der Jahre wurde eine Vielzahl von Methoden entwickelt, mit denen aus differentiellen SAR-Interferogrammen Deformationen der Erdoberfläche abgeleitet werden können. Hier wird zwischen der Gruppe der *Persistent Scatterer*-Methoden (PS) und der *Small Baseline Subset*-Methode (SBAS) nach BERARDINO ET AL. (2002) unterschieden.

Der erste Algorithmus im Bereich der PS-Anwendungen wurde von FERRETTI ET AL. (2001) formuliert und ist als *Permanent Scatterer*-Technik patentiert. Weitere bekannte Vertreter dieser Gruppen sind die *Stanford Method for Persistent Scatterers* (StaMPS) nach HOOPER ET AL. (2004); HOOPER (2006) und das *Spatio-Temporal Unwrapping Network* (STUN) von KAMPES UND ADAM (2005). All diese Methoden haben gemeinsam, dass sie für die Auswertung lediglich Punkte verwenden, die langzeitstabile Rückstreuungseigenschaften aufweisen, woraus sich auch der Name dieser Techniken ableiten lässt. Zudem werden nur Aufnahmen in ihrer ursprünglichen Auflösung verwendet, sogenannte *Single Look Complex*-Daten (SLC).

Für die Erzeugung der Interferogramme wird eine gemeinsame SAR-Szene gewählt, die als Master bezeichnet wird. Der so erzeugte Bildstapel wird daher oft auch *Single-Master-Stack* genannt. Dieses Vorgehen hat zwei Nachteile. Zum einen entstehen teilweise Interferogramme mit großen zeitlichen und örtlichen Basislinien und zum anderen wird bei N' SAR-Bildern die Anzahl der Interferogramme auf $N' - 1$ beschränkt. Dass ersteres einen Nachteil darstellt, haben ZEBKER UND VILLASENOR (1992) gezeigt. ZEBKER UND VILLASENOR (1992)

haben nachgewiesen, dass bei zunehmender (zeitlichen oder örtlichen) Basislinie die Kohärenz abnimmt. Dies ist auch der Grund, warum bei den PS-Techniken nur langzeitstabile Rückstreuer verwendet werden. Solche Punkte sind auch bei niedriger Kohärenz noch auswertbar.

Einen anderen Ansatz verfolgt die SBAS-Methode. Hier werden Interferogramme zwischen SAR-Aufnahmen gebildet, sodass die Basislinien bestimmte Schwellwerte nicht überschreiten. Auf diese Weise soll sichergestellt werden, dass die Kohärenz nicht zu stark abnimmt. Zur weiteren Rauschreduzierung wird eine sogenannte *Multilook*-Operation auf den Bildern ausgeführt. Unter der Annahme der statistischen Unabhängigkeit werden hierbei die Phasenwerte innerhalb eines definierten Fensters gemittelt. So entstehen jedoch Interferogramme mit niedrigerer Auflösung. Allerdings wird durch diese Wahl von Datenpaaren zu Interferogrammerzeugung auch die Anzahl der möglichen D-InSAR-Bilder gegenüber den PS-Methoden erhöht.

Auch zum Lösen der Phasenmehrdeutigkeiten gibt es eine große Anzahl von Ansätzen, die sich auf verschiedene Weisen gruppieren lassen. Eine mögliche Einteilung ist die Differenzierung zwischen 2D- und 3D-Verfahren. Die zweidimensionalen Methoden führen das Phase Unwrapping auf einzelnen Bildern aus. Hier lässt sich nochmals zwischen den *Branch-Cut*-Methoden (auch *Residue-Cut*- oder *Path-Following*-Methoden genannt), den Kleinst-Quadrat-Methoden und den *Network Flow*-Methoden unterscheiden (CHEN, 2001, S. 11ff). Die erste Formulierung eines Network Flow-Ansatzes geht auf COSTANTINI (1997) zurück, der das Lösen der Mehrdeutigkeiten auf einem vollbesetzten Interferogramm als Minimum Cost Flow (MCF)-Problem formuliert hat. Der Übergang auf dünnbesetzte Systeme wurde in COSTANTINI UND ROSEN (1999) vorgestellt.

Die bekanntesten Vertreter der 3D-Phase Unwrapping Methoden sind der *Extended Minimum Cost Flow* (EMCF)-Algorithmus von PEPE UND LANARI (2006) und der Edgelist-Algorithmus nach SHANKER UND ZEBKER (2010). Beide Verfahren gehören zu der Klasse der Network Flow-Methoden.

Für den EMCF-Algorithmus wird ein spezieller Bildstapel vorausgesetzt, der allerdings mit der SBAS-Methode kompatibel ist. Der Algorithmus ist in zwei Schritte unterteilt. Der erste Schritt wird als *Temporal Unwrapping* bezeichnet. Er basiert auf den zeitlichen und räumlichen Zusammenhängen, die zwischen den SAR-Aufnahmen und den Interferogrammen des Bildstapels existieren. Diese lassen sich mithilfe der Basislinien als Netzwerk darstellen, auf dem das zeitliche Phase Unwrapping durchgeführt wird. Dagegen wird der zweite Schritt, das *Spatial Unwrapping*, auf den einzelnen Interferogrammen ausgeführt. Er entspricht dem von COSTANTINI UND ROSEN (1999) formulierten Verfahren unter Verwendung der Ergebnisse aus dem Temporal Unwrapping.

Der Edgelist-Algorithmus ist für D-InSAR-Bildstapel anwendbar, die für PS-Techniken oder die SBAS-Methode erzeugt wurden. Zur Lösung der Mehrdeutigkeiten stellen die bisher genannten MCF-Ansätze Restriktionen für geschlossene Phasenzyklen auf. Hier macht der Edgelist-Algorithmus einen Übergang auf Restriktionen für jede Kante. Die Nutzung der Zeit als dritte Dimension wird dabei realisiert, indem Kanten zwischen Punkten von benachbarten Interferogrammen im Bildstapel eingefügt werden.

1.3 Wissenschaftliche Zielsetzung der Arbeit

In den letzten Jahren stieg die Anzahl der Methoden zur Auswertung von D-InSAR-Bildstapeln. Diese Verfahren verwenden dabei nur einen Teil der Bildpunkte, da in der Regel ein Großteil der Szenen aus verschiedenen Gründen nicht für die Auswertung nutzbar ist. Die Phasenabwicklung und Deformationsbestimmung mit D-InSAR-Bildstapeln basieren also auf der Ausnutzung mehrerer dünnbesetzter Systeme. Durch die Interferogrammerzeugung aus jeweils zwei SAR-Szenen bestehen zeitliche Zusammenhänge zwischen den Bildern im Stapel. Somit sind auch die dünnbesetzten Systeme miteinander verknüpft.

Ziel dieser Arbeit ist es folglich, die Zusammenhänge zu erarbeiten, die zwischen den kohärenten Pixeln

bestehen, die für eine Auswertung genutzt werden können. Außerdem gilt es, die Beziehung zwischen den Interferogrammen selbst zu erörtern. Die Verknüpfung zwischen all diesen Elementen soll in einer geeigneten Datenstruktur abgebildet werden. Zu diesem Zweck werden zwei Graphenstrukturen aufgebaut. Die erste dient der Darstellung der raum/zeitlichen Zusammenhänge zwischen den D-InSAR-Szenen des Bildstapels. In der zweiten Struktur werden die räumlichen Verknüpfungen zwischen den kohärenten Pixeln abgebildet. Schließlich sind die beiden Graphen noch geeignet miteinander zu verbinden.

Des Weiteren sollen geeignete Anfragen an die Datenstruktur formuliert werden, um die Untersuchung und Bearbeitung des Bildstapels zu ermöglichen. Diese sollen auch die Nutzbarkeit der Datenstruktur für weitere Prozessierungsschritte realisieren.

1.4 Gliederung der Arbeit

Zu Beginn dieser Arbeit werden in Kapitel 2 die Grundlagen zur Beschreibung eines D-InSAR-Bildstapels erläutert. Dafür wird erklärt, was unter einem differentiellen SAR-Interferogramm zu verstehen ist und wofür es verwendet werden kann. Anschließend wird die Kohärenz als qualitätsbestimmendes Merkmal eingeführt und gezeigt, welchen Einfluss sie auf eine mögliche Auswertbarkeit von Pixeln hat. Des Weiteren wird beschrieben, welche Auswirkungen verschiedene Anwendungen zur Deformationsanalyse bzw. für das Phase Unwrapping auf die Struktur des Bildstapels haben. Dazu werden für die verschiedenen Bildstapel die unterschiedlichen Graphen in der Basislinienebene dargestellt. Abschließend wird die Datengrundlage dieser Arbeit vorgestellt und einer der vorher erläuterten Strukturen zugeordnet.

In Kapitel 3 werden einige Anwendungen vorgestellt, die als weitere Prozessierungsschritte für den gegebenen D-InSAR-Bildstapel in Frage kommen. Hierfür werden zunächst die allgemeinen Grundlagen des Phase Unwrappings erläutert, da das Lösen der Phasenmehrdeutigkeiten der nächste Schritt in der Weiterverarbeitung eines D-InSAR-Bildstapels ist. Im Anschluss wird die Phasenabwicklung als MCF-Problem formuliert. Diese Formulierung wird zur Vorstellung zweier spezieller Phase Unwrapping Verfahren benötigt, den EMCF-Algorithmus von PEPE UND LANARI (2006) und den Edgelist-Algorithmus nach SHANKER UND ZEBKER (2010). Zum Schluss wird mit der SBAS-Methode nach BERARDINO ET AL. (2002) eine Anwendung vorgestellt, mit der aus einem Bildstapel von abgewickelten D-InSAR-Szenen Deformationen der Erdoberfläche bestimmt werden können.

Aus diesen theoretischen Grundlagen werden in Kapitel 4 Anforderungen an die Datenstruktur abgeleitet. Anschließend wird das Modell der entwickelten Datenstruktur mit Hilfe von UML-Diagrammen dokumentiert. Ergänzt wird dies durch die Vorstellung von Anfragen, die von der Datenstruktur unterstützt werden. In Kapitel 5 wird auf die Implementierung der Datenstruktur eingegangen. Dabei wird die Wahl der Programmiersprache begründet. Außerdem wird ein Testdatensatz vorgestellt, mit dem die Lauffähigkeit des Programmpakets geprüft werden soll. Dies wird durch ein Funktionshandbuch im Anhang C nachgewiesen. Abschließend wird in Kapitel 6 die hier präsentierte Arbeit resümiert. Ausstehende und anknüpfende Aufgaben werden in einem Ausblick zusammengefasst.

2. Grundlagen für die Beschreibung von D-InSAR-Bildstapeln

Wie im vorangegangenen Kapitel beschrieben, besteht das Ziel dieser Arbeit darin Datenstrukturen zu entwickeln, die die Zusammenhänge zwischen kohärenten Pixeln in D-InSAR-Bildstapeln darstellen. Diese Datenstrukturen werden in der weiteren Prozessierung insbesondere beim Lösen der Phaseneindeutigkeiten benötigt. Hierfür werden nun die Grundlagen zur Beschreibung dieser Daten erarbeitet.

Um ein Verständnis dafür zu entwickeln, worum es sich bei D-InSAR-Bildern handelt und welche Informationen sie enthalten, wird kurz der Entstehungsweg skizziert. Begonnen wird bei den SAR-Szenen in Abschnitt 2.1, die die Ausgangsdaten in der Prozessierung sind. Im Anschluss wird die Radarinterferometrie erläutert (Abschnitt 2.2), was auf die differentielle Radarinterferometrie führt (Abschnitt 2.3).

Im darauffolgenden Abschnitt 2.4 wird das Ähnlichkeitsmaß der Kohärenz vorgestellt. Aufgrund dessen kann begründet werden, warum nur kohärente Pixel für eine weitere Auswertung und damit in der Datenstruktur berücksichtigt werden. Abschließend wird in Abschnitt 2.5 auf die Erzeugung von D-InSAR-Bildstapeln eingegangen. Hier wird im Speziellen erläutert, wie Datenpaare zur Interferogrammerzeugung gewählt werden und welchen Einfluss dies auf die Struktur des Stapels hat (Abschnitt 2.5.1). Außerdem wird die Datengrundlage für diese Arbeit vorgestellt und in die vorangegangene Beschreibung von Bildstapelstrukturen eingegliedert (Abschnitt 2.5.2).

2.1 Radar mit synthetischer Apertur

In diesem Abschnitt soll auf die relevanten Grundlagen von Radarsystemen mit synthetischer Apertur (engl. *Synthetic Aperture Radar*, SAR) eingegangen werden. Dabei wird bewusst auf die physikalisch-mathematische Beschreibung des Aufnahmeverfahrens verzichtet und lediglich auf geeignete Literatur verwiesen, in der die Funktionsweise von SAR-Systemen erläutert wird. Einen guten Einstieg bieten SEIFERT UND ZINK (1993), die den Messablauf eines Radarsystems physikalisch beschreiben, beginnend bei Radaren mit realer Apertur (RAR) hin zu den Dopplermessungen von SAR-Systemen. Eine umfassende Darstellung relevanter Eigenschaften von SAR-Aufnahmen in Hinblick auf die weitere Verwendung für Radarinterferometrie ist in HANSEN (2001) gegeben. Für tiefer gehende Studien bietet sich KLAUSING UND HOLPP (2000) an.

In dieser Arbeit wird SAR als ein bildgebendes, satellitengestütztes System aufgefasst, das auf Messungen mit Mikrowellenstrahlung im C-Band (Wellenlänge $\lambda = 5,66\text{cm}$) beruht, da die Datengrundlage aus solchen Aufnahmen der ERS-1/2 Satelliten erzeugt wurde. Bei SAR handelt es sich prinzipiell um eine spezielle Aufnahmetechnik, die die Auflösung in Flugrichtung (oftmals als Azimuthrichtung bezeichnet) eines Radarbildes erhöhen soll. Bei den klassischen Radaren mit realer Apertur ist die Auflösung in Azimuthrichtung zum einen entfernungsabhängig und zum anderen abhängig vom Verhältnis der verwendeten Wellenlänge zur Antennenlänge. Bei satellitengestützten Radaren würde dies zu unpraktikablen Antennenlängen führen, um eine nutzbare Auflösung zu erhalten (SEIFERT UND ZINK, 1993).

Im Gegensatz dazu misst ein Radar mit synthetischer Apertur neben der Intensität des zurückgestreuten Signals auch dessen Phase oder genauer das Phasenreststück der ankommenden Welle. Anhand der zusätzlichen Phasenmessung kann bei geeigneter Auswertung der Doppler-Verschiebungen, die durch die Bewegung des Satelliten zwischen Aussenden und Empfangen des Signals entsteht, das Auflösungsvermögen in Flugrichtung erhöht werden. Dieses entspricht bei SAR-Systemen der halben Länge der Apertur, womit bildgebende

Radarmessungen vom Satelliten aus möglich werden.

Aus einer SAR-Aufnahme lassen sich somit zwei Bilder erzeugen, ein Intensitätsbild und ein Phasenbild. Während ersteres ein Grauwertbild ist und in der klassischen Fernerkundung wie eine optische Aufnahme verwendet wird, bildet das Phasenbild die Grundlage für die Radarinterferometrie.

2.2 Radarinterferometrie

Die Radarinterferometrie, oftmals InSAR (für *Interferometric Synthetic Aperture Radar*) genannt, basiert auf der Auswertung zweier Phasenbilder der gleichen Region, die von leicht unterschiedlichen Positionen aufgenommen werden. Dadurch ist es möglich, aus zweidimensionalen Aufnahmen Informationen über die Geländehöhe zu extrahieren, ähnlich der Tiefenwahrnehmung beim stereoskopischen Sehen (ROSEN ET AL., 2000). Es wird aufgrund der Aufnahmekonstellationen zwischen der *Along-Track*- und der *Across-Track-Interferometrie* unterschieden. Bei der *Across-Track-Interferometrie* wird nochmals zwischen der *Single-Pass*- und *Two-Pass-Interferometrie* (auch *Repeat-Pass-Interferometrie*) differenziert. Allerdings soll hier lediglich die *Two-Pass-Interferometrie* vorgestellt werden, bei der die SAR-Aufnahmen von leicht versetzten Satellitenorbits zu zwei verschiedenen Zeitpunkten erstellt werden. Bei den vorliegenden Daten ist dies der Fall. Die Hauptanwendung dieser Technik ist die Erzeugung von *Digitalen Höhenmodellen* (DHM) (SÖRGEL, 2006). Prinzipiell entspricht die interferometrische Phase $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ eines Pixels $x(i,j)$ der Differenz der Phasenwerte $\phi_{x(i,j)}^{t_{\alpha}}$ und $\phi_{x(i,j)}^{t_{\beta}}$ der beiden SAR-Aufnahmen zu den Zeitpunkten t_{α} und t_{β}

$$\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}} = \phi_{x(i,j)}^{t_{\beta}} - \phi_{x(i,j)}^{t_{\alpha}}, \quad (2.1)$$

wobei die Aufnahme zum Zeitpunkt t_{α} als Masterszene und die zum Zeitpunkt t_{β} als Slaveszene bezeichnet wird. Voraussetzung hierfür ist, dass beide Aufnahmen im Subpixelbereich auf das gleiche Gitter registriert sind, was durch die sogenannte Koregistrierung ermöglicht wird (HANSEN, 2001, S. 45ff).

Neben der Information zur Höhe enthält die interferometrische Phase $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ auch Anteile aus anderen Quellen. Dies lässt sich nach HALSIG (2012, S. 44) als additives Modell darstellen:

$$\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}} = \phi_{x(i,j)|topo}^{\Delta t_{\alpha,\beta}} + \phi_{x(i,j)|defo}^{\Delta t_{\alpha,\beta}} + \phi_{x(i,j)|flatearth}^{\Delta t_{\alpha,\beta}} + \phi_{x(i,j)|atmo}^{\Delta t_{\alpha,\beta}} + \phi_{x(i,j)|noise}^{\Delta t_{\alpha,\beta}}. \quad (2.2)$$

Der entscheidende Parameter bei der SAR-Interferometrie ist $\phi_{x(i,j)|topo}^{\Delta t_{\alpha,\beta}}$, der für die Bestimmung der Geländehöhe verwendet wird. Durch Bewegungen der Erdoberfläche entsteht der Phasenanteil $\phi_{x(i,j)|defo}^{\Delta t_{\alpha,\beta}}$, der die Messgröße bei der differentiellen SAR-Interferometrie darstellt, wie in Abschnitt 2.3 erläutert wird. Der Phasenanteil der „flachen Erde“ (engl. *flat earth*) $\phi_{x(i,j)|flatearth}^{\Delta t_{\alpha,\beta}}$ wird durch die Aufnahmegeometrie hervorgerufen. SAR-Szenen werden in Schrägsicht aufgenommen, daher kommt es durch die unterschiedlichen Aufnahmepositionen auch bei vollkommen ebener Erdoberfläche zu regelmäßigen Phasentrends (SCHÄFER, 2012, S. 42). Die SAR-Aufnahmen, die ein Interferogramm bilden, werden zu unterschiedlichen Zeitpunkten aufgenommen. Aus diesem Grund ist davon auszugehen, dass sich die atmosphärischen Begebenheiten nicht gleichen. Dies hat zur Folge, dass die Ausbreitungscharakteristik der Radarwelle für beide Zeitpunkte verschieden ist. Diesem Effekt wird durch den Anteil $\phi_{x(i,j)|atmo}^{\Delta t_{\alpha,\beta}}$ Rechnung getragen. Der letzte Summand $\phi_{x(i,j)|noise}^{\Delta t_{\alpha,\beta}}$ fasst das Phasenrauschen zusammen, das auf diverse Ursachen zurückgeführt werden kann. Dieser Anteil steht in engem Zusammenhang mit der Kohärenz, die in Abschnitt 2.4 detaillierter betrachtet wird. Nach SHANKER (2010, S. 10–11) stellen für diese Anwendung alle Anteile außer $\phi_{x(i,j)|topo}^{\Delta t_{\alpha,\beta}}$ und $\phi_{x(i,j)|flatearth}^{\Delta t_{\alpha,\beta}}$ zufällige Abweichungen dar. Der Phasenanteil der flachen Erde ist eine systematische Abweichung, die unter Verwendung präziser Orbits eliminiert werden kann.

Das Rauschen $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}|_{noise}$ lässt sich hingegen durch eine *Multilook* Operation reduzieren. Dabei werden die Phasen in einem vorher definierten Fenster gemittelt, was allerdings zu einer Reduktion der Auflösung führt. Bei ERS-Szenen wird für das Fenster in der Regel ein Verhältnis von 1 : 5 in Range- bzw. Azimuthrichtung gewählt, da dies zu annähernd quadratischen Pixeln führt (HANSEN, 2001, S. 52). Die Anzahl der Pixel, über die gemittelt wird, wird oftmals als *Looks* bezeichnet. Wählt man also ein 4×20 großes Fenster, ergeben sich 80 Looks. In diesem Zusammenhang wird häufig von 80-Look-Bildern gesprochen.

Eine weitere wichtige Eigenschaft der interferometrischen Phase ist dadurch bedingt, dass ein SAR-System nur Phasenreststücke detektiert. Aus diesem Grund kann die interferometrische Phase ebenfalls nur in einem Wertebereich von $[-\pi, \pi)$ bestimmt werden. Um Höheninformationen zu erhalten, müssen diese mehrdeutigen Phasen in absolute Werte überführt werden. Dieser Prozess wird als *Phase Unwrapping* bezeichnet und in Kapitel 3 behandelt.

Für eine ausführliche Beschreibung der Prozessierung von InSAR-Daten sei auf HANSEN (2001) verwiesen. Eine umfangreiche Einführung in die systemtheoretischen Grundlagen der Radarinterferometrie bieten beispielsweise ROSEN ET AL. (2000) und MADSEN UND ZEBKER (1998).

2.3 Differentielle Radarinterferometrie

Im vorangegangenen Abschnitt wurde mit $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}|_{defo}$ in Formel (2.2) der Phasenanteil der interferometrischen Phase vorgestellt, der durch Deformationen der Erdoberfläche zwischen den zwei Aufnahmezeitpunkten t_α und t_β bedingt ist. Dieser Anteil bildet die Messgröße für die differentielle Radarinterferometrie (D-InSAR), mit der es prinzipiell möglich ist, Deformationen im Millimeter- bis Zentimeterbereich zu detektieren (CASU, 2009, S. 34 ff).

Um dies zu realisieren, muss aus dem Interferogramm der Anteil für die Topographie $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}|_{topo}$ herausgerechnet werden. Dazu wird ein DHM herangezogen, das in ein Phasenbild umgerechnet und von der interferometrischen Phase abgezogen wird. Daher wird diese Technik auch als differentielle SAR-Interferometrie bezeichnet. Das verwendete Höhenmodell kann entweder aus einer externen Quelle stammen oder aus einem Interferogramm abgeleitet sein. Nach Abzug der topographischen Phase und des Anteils der flachen Erde, setzt sich nach CASU (2009, S. 34 ff) der Phasenwert eines Pixels $x(i,j)$ in einem D-InSAR Bild aus

$$\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}} = \phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}|_{defo} + \phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}|_{\Delta topo} + \phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}|_{orbit} + \phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}|_{atmo} + \phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}|_{noise} \quad (2.3)$$

zusammen. Abweichend von Formel (2.2) ist nun der Phasenanteil $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}|_{\Delta topo}$ enthalten, der topographischer Fehler genannt wird und durch Fehler im verwendeten DHM bzw. bei der Umrechnung in ein Phasenbild bedingt ist. Eine weitere differentielle Abweichung ist mit $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}|_{orbit}$ gegeben, die oftmals als Orbitfehler bezeichnet wird. Dieser entsteht bei der Elimination des Phasenanteils der flachen Erde und ist auf Ungenauigkeiten in den zur Korrektur verwendeten Bahndaten zurückzuführen.

Analog zur interferometrischen Phase im vorherigen Abschnitt ist auch die differentielle, interferometrische Phase ein mehrdeutiges Signal im Wertebereich von $[-\pi, \pi)$. Für die Berechnung der Oberflächenbewegung ist es auch hier notwendig, die absoluten Phasenwerte mittels Phase Unwrappings zu bestimmen.

2.4 Kohärenz

Wie in den vorherigen Abschnitten beschrieben, basiert die (differentielle) Radarinterferometrie auf der Auswertung zweier Phasenmessungen des gleichen Aufnahmegebietes von verschiedenen Positionen. Des Weiteren

wurden die D-InSAR-Bilder für diese Arbeit mit der Technik der Two-Pass-Interferometrie erzeugt; es liegen also verschiedene Aufnahmezeitpunkte für die SAR-Szenen eines Interferogramms vor. Im Folgenden soll beschrieben werden, wie sich die räumlichen und zeitlichen Unterschiede bei der Aufnahme auf die Qualität der erzeugten D-InSAR-Szenen auswirken können, oder anders ausgedrückt, wie dadurch die Genauigkeit der interferometrischen Phase beeinflusst wird.

LEE ET AL. (1994) haben einen analytischen Ausdruck gefunden, um die bedingte Wahrscheinlichkeitsdichtefunktion von mehrdeutigen Multilook-Phasenwerten in Abhängigkeit der Kohärenz γ auszudrücken. Der Begriff der Kohärenz, wie er hier aufgegriffen wird, entspricht dem Betrag des komplexen Korrelationskoeffizienten γ_c der beiden SAR-Signale s_1 und s_2 , der definiert ist über

$$\gamma_c = \frac{E\{S_1 S_2^*\}}{\sqrt{E\{|S_1|^2\}} \sqrt{E\{|S_2|^2\}}}, \quad (2.4)$$

wobei $(.)^*$ für die komplexe Konjugation steht (CASU, 2009, S. 26). Die Kohärenz $\gamma = |\gamma_c|$ kann daher zum einen als ein Ähnlichkeitsmaß für die Phasensignale der beiden SAR-Messungen aufgefasst und zum anderen für die Ableitung der Genauigkeit der interferometrischen Phase verwendet werden (LEE ET AL., 1994).

Unter Annahme von Ergodizität lässt sich die Kohärenz für ein Interferogramm schätzen, indem die Erwartungswerte durch ein räumliches Mittel über ein Bildfenster ersetzt wird. Daraus ergibt sich der gängige Kohärenzschätzer als ein gleitendes Mittel:

$$\hat{\gamma} = \left| \frac{\sum_{k=1}^N s_{1,k} s_{2,k}^*}{\sqrt{\sum_{k=1}^N |s_{1,k}|^2} \sqrt{\sum_{k=1}^N |s_{2,k}|^2}} \right|. \quad (2.5)$$

Die Regionen, über die gemittelt wird, werden dabei wie bei der Multilook Operation zur Rauschreduktion festgelegt, sodass N hier wieder der Anzahl der Looks entspricht und die Auflösungszellen des entstehenden Kohärenzbilds mit den Pixeln des korrespondierenden Interferogramms übereinstimmen.

Die Kohärenz nimmt einen Wertebereich $[0, 1]$ an. Laut DE SOUSA (2009, S. 35) bedeutet eine Kohärenz von $\gamma = 1$, dass das zurückgestreute Signal in beiden Bildern identisch ist und die interferometrische Phase als fehlerfrei angenommen werden kann (LEE ET AL., 1994). Hingegen folgt aus einem Wert von $\gamma = 0$, dass die beiden Signale keine gemeinsame Information enthalten. Man spricht hier von Unkorreliertheit oder vollständiger Dekorrelation. Entsprechend wird die Abnahme der Kohärenz als Dekorrelation bezeichnet.

Für den Dekorrelationseffekt gibt es zahlreiche Einflussfaktoren, die sich teilweise in unterschiedliche Gruppen einteilen und somit in verschiedenen Kombinationen darstellen lassen. ZEBKER UND VILLASENOR (1992) haben gezeigt, dass sich die Gesamtkohärenz als Produkt der Kohärenzfaktoren für die verschiedenen Dekorrelationseffekte ergibt. So lässt sich die Kohärenz beispielsweise in

$$\gamma = \gamma_{räumlich} \cdot \gamma_{zeitlich} \cdot \gamma_{Doppler} \cdot \gamma_{andere} \quad (2.6)$$

aufteilen. Hier wird sich auf die Einflüsse beschränkt, die im weiteren Verlauf der Arbeit (vgl. Abschnitt 2.5) verwendet werden. Alle anderen Effekte wie das thermische Rauschen oder Dekorrelation durch Prozessierungsfehler (bspw. in der Koregistrierung) werden zu γ_{andere} zusammengefasst.

Nachfolgend werden die relevanten Effekte kurz beschrieben und mit den vorliegenden Daten visuell verdeutlicht.

Räumliche Dekorrelation:

Unter dem Begriff der räumlichen Dekorrelation werden in der Regel zwei Effekte zusammengefasst, die mit der Aufnahmegeometrie durch die unterschiedlichen Orbitpositionen zusammenhängen. Zum einen ist hier der Einfluss durch die scheinbare Rotation von Rückstreuern in einer Auflösungszelle durch Nicht-

Parallelität der Orbits zu nennen und zum anderen die Dekorrelation durch verschiedene Blickwinkel (bzw. Einfallswinkel) auf die Rückstreuer.

Laut ZEBKER UND VILLASENOR (1992) führt bereits eine Abweichung der Orbits um einen Winkel von $0,7^\circ$ zur vollständigen Dekorrelation bei der Verwendung eines C-Band-Radarsystems, wie es die ERS-Satelliten verwenden. Allerdings kann der Effekt nach KNABE (2004, S. 37) aufgrund der Lagestabilität von ERS-1/-2 vernachlässigt werden.

Kritischer ist der Effekt durch die Blickwinkelunterschiede. Dieser lässt sich als Funktion der Basislinie darstellen, also in Abhängigkeit der Distanz zwischen den beiden Satellitenpositionen. Oftmals wird hier auch die Abhängigkeit von der orthogonalen Basislinie verwendet, die der Projektion auf die Richtung entspricht, die orthogonal auf der Blickrichtung des Masters liegt. Aus dieser Funktion lässt sich ein Wert ableiten, für den vollständige Dekorrelation (bei ERS-Interferogrammen) erwartet wird. Dieser Wert wird oftmals als *kritische Basislinie* bezeichnet und liegt je nach Quelle zwischen $1100m$ und $1200m$. Jedoch sind bereits Interferogramme mit einer orthogonalen Basislinie über $600m$ sehr schwer auszuwerten (CASU, 2009, S. 31). Für eine genauere, vor allem mathematisch-physikalische Beschreibung der räumlichen Dekorrelation sei hier insbesondere auf (ZEBKER UND VILLASENOR, 1992) verwiesen.

Zeitliche Dekorrelation:

Die zweite Gruppe von Dekorrelationseffekten $\gamma_{zeitlich}$ fasst all die Einflüsse zusammen, welche die Rückstreuungseigenschaften in einer Auflösungszelle zwischen den Aufnahmezeitpunkten der SAR-Signale s_1 und s_2 verändern. Dies kann zum einen auf atmosphärische Unterschiede zurückgeführt werden. Hier wird nochmals zwischen Effekten differenziert, die die Erdoberfläche betreffen (Bodenfeuchte, Schneebedeckung usw.) sowie zwischen Unterschieden in der Signalausbreitung in der Atmosphäre aufgrund der Wetterlage und sich dadurch ändernden optischen Dichten (KNABE, 2004, S. 39; CASU, 2009, S. 33).

Zum anderen umfasst $\gamma_{zeitlich}$ die Dekorrelation durch die Änderung des Streuvolumens. Darunter wird die geometrische Änderung der einzelnen Rückstreuer in einer Auflösungszelle verstanden. Dies kann sowohl natürliche (z.B. Wachstum von Vegetation) als auch anthropogene (z.B. durch Ackerbau) Ursachen haben. Nicht eingeschlossen sind Phänomene, bei denen sich alle Rückstreuer in eine gemeinsame Richtung bewegen, wie es bei langsamen Deformationsprozessen geschieht. Dies führt zu einer systematischen Phasenverschiebung statt zu einer Dekorrelation und stellt die Messgröße für die differentielle Radarinterferometrie dar, wie in Abschnitt 2.3 beschrieben (ZEBKER UND VILLASENOR, 1992).

All diese Dekorrelationseffekte ereignen sich je nach Oberflächentypen in verschiedenen Zeitskalen, welche sich über Sekundenbruchteile bis über Jahre hinweg erstrecken (MADSEN UND ZEBKER, 1998, S. 22; CASU, 2009, S. 33). Die zeitliche Dekorrelation ist ein limitierender Faktor der Two-Pass-Interferometrie, da in aller Regel nur ein Teil der Aufnahme auswertbar ist. Meistens beruht die Auswertung auf Regionen, die eine langsame zeitliche Dekorrelation erfahren, wozu hauptsächlich urbane Gebiete zählen. Hier kann auch bei zeitlichen Basislinien von mehreren Jahren mit einer hohen Kohärenz gerechnet werden.

Dekorrelation durch die Differenz der Doppler-Frequenzen:

Der letzte Faktor, der die Kohärenz beeinflusst und hier kurz erläutert werden soll, ist die Differenz der mittleren Doppler-Frequenzen (engl. *Doppler Centroid Frequency*, oft nur *Doppler Centroid*) der beiden SAR-Szenen. Für eine genaue Beschreibung des Doppler-Zentroids und verschiedene Möglichkeiten seiner Bestimmung sei auf MADSEN (1989), WONG UND CUMMING (1996) und YANNAM ET AL. (2012) verwiesen. Es sei jedoch erwähnt, dass es sich bei der mittleren Doppler-Frequenz um einen entscheidenden Parameter zur Prozessierung der Rohdaten eines SAR-Systems in Radaraufnahmen handelt.

Bei zunehmender Differenz der mittleren Doppler-Frequenz von zwei SAR-Aufnahmen nimmt der gemeinsame Informationsgehalt ihrer Phasensignale ab. Dies führt bei der Erzeugung eines Interferogramms aus diesen

beiden Szenen zu einer geringen Kohärenz und damit zu einer verstärkten Dekorrelation der Phasenwerte (CASU, 2009, S. 31–32; DE SOUSA, 2009, S. 40).

Visuelle Darstellung von Dekorrelationseffekten:

In Abbildung 2.1 sind die beschriebenen Dekorrelationseffekte anhand von Kohärenz- und Phasenbildern dargestellt. Die Abbildungen umfassen das Stadtgebiet von Köln und seine Umgebung, sodass sowohl urbane als auch ländliche Regionen gezeigt werden.

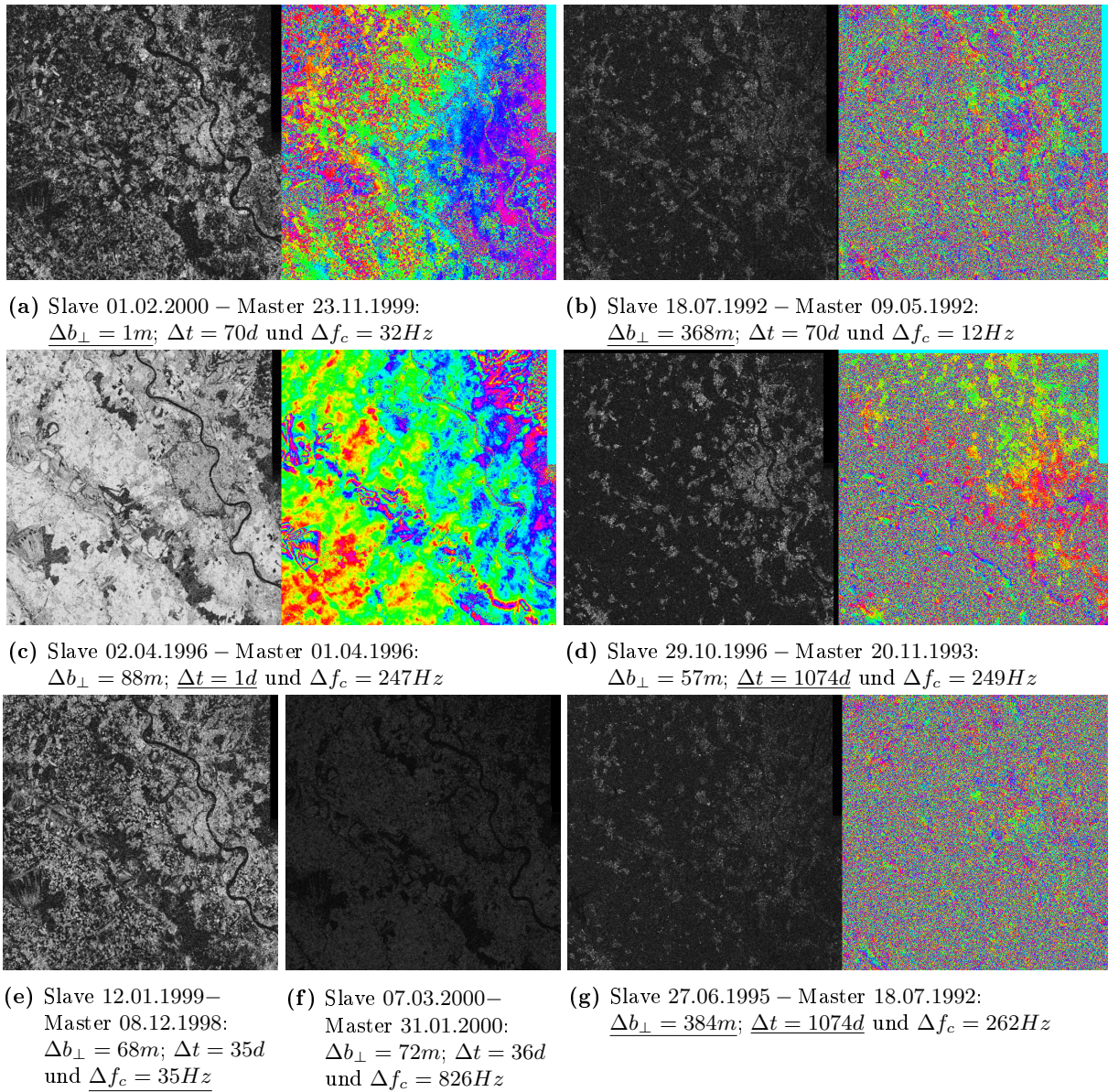


Abbildung 2.1: Einfluss verschiedener Dekorrelationseffekte auf interferometrische Bilddaten. Die obere Reihe zeigt den Einfluss einer größer werdenden orthogonalen Basislinie; die zweite Reihe den Einfluss einer größer werdenden zeitlichen Basislinie; in der letzten Reihe sind der Kohärenzverlust durch zunehmende Doppler-Frequenz-Differenz (e) und (f) sowie durch eine große räumliche und zeitliche Basislinie (g) dargestellt. Die Grauwertbilder sind Kohärenzbilder, wobei weiß für $\gamma = 1$ und schwarz für $\gamma = 0$ steht, und die RGB-Bilder sind die korrespondierenden D-InSAR Szenen.

Es wird für jeden Parameter (orthogonale Basislinie Δb_{\perp} , zeitliche Basislinie Δt und Doppler-Frequenz-Differenz Δf_c) ein Interferogrammpaar gesucht, sodass sich der zu untersuchende Parameter (in der Bild-

unterschrift die unterstrichene Größe) stark ändert und die anderen Parameter nahezu gleich bleiben. Eine Ausnahme bildet hier Abbildung 2.1g, die den kombinierten Effekt einer großen räumlichen (Abb. 2.1b) und zeitlichen (Abb. 2.1d) Basislinie zeigen soll. Die zusätzliche Darstellung der Interferogramme verdeutlicht den am Anfang des Abschnitts 2.4 beschriebenen Zusammenhang zwischen der Kohärenz und dem Phasenrauschen.

In den Abbildungen ist deutlich zu erkennen, dass in Gebieten mit niedriger Kohärenz (in dunkelgrau bis schwarz dargestellt) in den korrespondierenden Phasenbildern keine zusammenhängenden Phaseninformationen vorhanden sind, sondern ein zufälliges Rauschen der Phasenwerte vorherrscht. Hingegen ist bei hohen Kohärenzwerten (hellgrau bis weiß) eine zusammenhängende Fläche in den Phasenbildern zu sehen. Dies bestätigt, dass bei der Auswertung von differentiellen SAR-Interferogrammen nur Pixel mit hoher Kohärenz herangezogen werden sollen.

2.5 Erzeugung eines Bildstapels von differentiellen SAR-Interferogrammen

Wie bereits in Kapitel 1 dargelegt, ist es von großem Interesse eine Deformationszeitreihe aus Radardaten zu erzeugen. In den vorangehenden Abschnitten wurden die Grundlagen der differentiellen SAR-Interferometrie erläutert. Mit Hilfe der D-InSAR-Daten ist es möglich, Deformationen im Zentimeter- bis Millimeterbereich zwischen zwei Aufnahmezeitpunkten zu detektieren. Um eine Zeitreihe aufzubauen, ist es notwendig eine geeignete Anzahl von Interferogrammen zu erzeugen, also einen Bildstapel zu generieren. Wie die Wahl von Aufnahmepaaren zum Erstellen von D-InSAR-Bildern getroffen wird, hängt dabei maßgeblich von der Methode ab, mit der die Deformationsinformationen extrahiert werden sollen.

Hier wird im Allgemeinen zwischen zwei Herangehensweisen unterschieden. Die erste Gruppe von Techniken zur Deformationsanalyse basiert auf der Verwendung von stabilen Rückstreuern (engl. *Persistent Scatterers*, kurz PS). Dies sind Punkte, die auch bei großen zeitlichen und räumlichen Basislinien ein stabiles Rückstreuerverhalten aufweisen. Als erstes wurde diese Idee von FERRETTI ET AL. (2001) in ihrem *Permanent Scatterer*-Algorithmus formuliert. Diese Grundidee wurde aufgegriffen und in anderen Algorithmen umgesetzt. Hier seien beispielsweise die bekannten Methoden *Stanford Method for Persistent Scatterers* (StaMPS) (HOOPER ET AL., 2004; HOOPER, 2006) und *Spatio-Temporal Unwrapping Network* (STUN) (KAMPES UND ADAM, 2005) genannt.

Der zweite Ansatz beruht darauf, den Dekorrelationseffekt möglichst klein zu halten, indem nur solche Paare gebildet werden, die kleine Basislinien aufweisen (vgl. Abschnitt 2.4). Bekanntester Repräsentant dieser Art von Methoden ist der *Small Baseline Subset*-Algorithmus (SBAS) von BERARDINO ET AL. (2002), der in Abschnitt 3.5 genauer vorgestellt wird.

Neben diesen Methoden zur Bestimmung von Deformationen gibt es auch Phase Unwrapping-Anwendungen, die spezielle Bildstapel voraussetzen. Hier sei insbesondere der *Extended Minimum Cost Flow*-Algorithmus (EMCF) von PEPE UND LANARI (2006) erwähnt, da dieser maßgeblich die Datenstrukturen beeinflusst hat, die in dieser Arbeit entwickelt werden (vgl. Abschnitt 3.3 und Kapitel 4).

Im Folgenden Abschnitt 2.5.1 wird erläutert, welche Anforderungen diese zwei Ansätze zu Deformationsanalyse und der EMCF-Algorithmus an den Datenstapel stellen. Hierbei wird vor allem auf die Möglichkeiten zur Datenpaar-Selektion eingegangen. Außerdem wird gezeigt, dass sich die Zusammenhänge zwischen den SAR-Aufnahmen und den Interferogrammen als Graph darstellen lassen. Im Anschluss wird die Datengrundlage dieser Arbeit beschrieben und darauf eingegangen, wie sich diese in die zuvor beschriebenen Arten von Bildstapeln einordnen lässt (vgl. Abschnitt 2.5.2).

2.5.1 Methoden zur Selektion von Datenpaaren

Wie im vorangehenden Abschnitt beschrieben, hängt die Wahl der Datenpaare zur Bildung eines Stapels von differentiellen SAR-Interferogrammen maßgeblich davon ab, welche Technik zur weiteren Datenprozessierung bzw. -analyse verwendet werden soll.

Bildstapel bei den Persistent Scatterer-Methoden:

Die Algorithmen die zur Klasse der PS-Techniken gehören, arbeiten ausnahmslos auf Daten in voller Auflösung, sogenannten *Single-Look*-Bildern. Weiterhin wird eine gemeinsame Masterszene für alle Interferogramme verwendet und somit ein *Single-Master*-Stapel aufgebaut. Dabei wird die Masterszene so ausgewählt, dass die maximal auftretende orthogonale und zeitliche Basislinie minimiert werden und somit die Dekorrelation klein gehalten wird (FERRETTI ET AL., 2001; HOOPER ET AL., 2004; KAMPES UND ADAM, 2005). Zusätzlich wird aus selbigen Grund versucht, die maximale Doppler-Frequenz-Verschiebung zu begrenzen (vgl. Abschnitt 2.4).

Angenommen es stehen N' SAR-Aufnahmen zur Verfügung und alle Bilder wurden bezüglich der gewählten Masterszene koregistriert, so lässt sich für jede Aufnahme eine zeitliche Basislinie $\Delta t_\alpha = t_\alpha - t_{Master}$ mit $\alpha = 1 \dots N'$ bestimmen. Dabei steht t_α für den Aufnahmezeitpunkt der α -ten Szene und t_{Master} für den Aufnahmezeitpunkt der Referenzszene. Außerdem lässt sich für jede Szene eine orthogonale Basislinienkomponente $b_{\perp|\alpha}$ bezüglich der gewählten Referenz berechnen. Wird nun eine Ebene mit den Basislinien ($T \times B_{\perp}$ -Ebene) aufgespannt, können die SAR-Aufnahmen als Punkte $P_\alpha = (\Delta t_i, b_{\perp i})$ in dieser Ebene dargestellt werden. Weiterhin lassen sich nun die erzeugten Interferogramme als Kanten zwischen den Punkten $P_{Master} = (0,0)$, der mit der Masterszene korrespondiert, und dem Punkt P_α darstellen, welcher der Slaveszene entspricht. Wird ein Single-Master-Stapel, wie er für PS-Methoden verwendet wird, auf die gerade beschriebene Weise in die Basislinienebene übertragen, entsteht ein Stern-Netzwerk mit $N' - 1$ Interferogrammen (vgl. Abbildung 2.2).

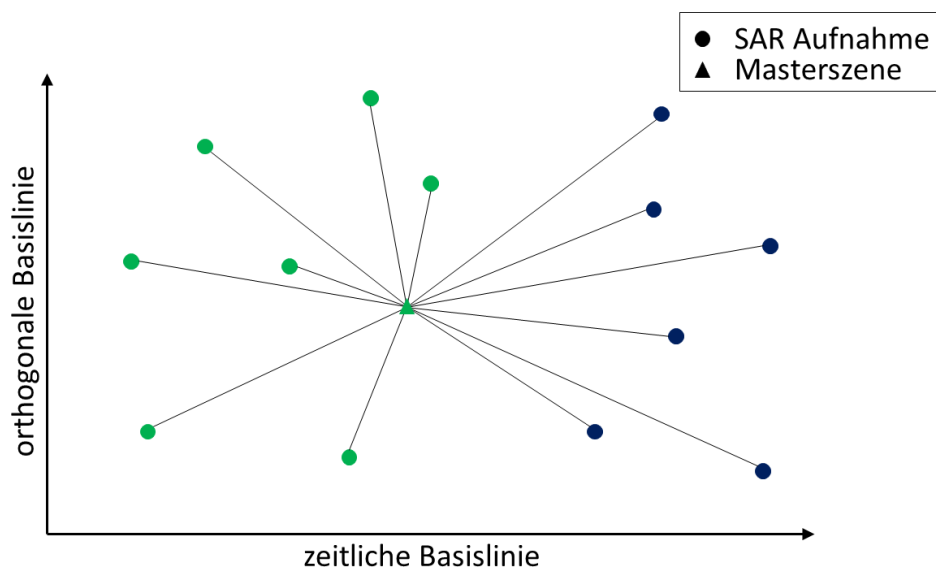


Abbildung 2.2: Basislinienetzwerk eines Single-Master-Stapels, wie er für PS-Methoden verwendet wird.

Zwar werden bei der Wahl der Masterszene die maximale zeitliche und räumliche Basislinie minimiert, doch es können dennoch große Basislinie auftreten. Deshalb verwenden PS-Methoden nur Pixel mit stabilen Rückstreueigenschaften, um der zunehmenden Dekorrelation entgegenzuwirken.

Bildstapel bei der SBAS-Methode:

Zur Reduzierung von Dekorrelationseffekten können alternativ Bildstapel mit Interferogrammen verwendet werden, die nur kleine Basislinien aufweisen. Eine solche Herangehensweise wird von der SBAS-Methode realisiert (CASU, 2009, S. 52ff).

Auch bei diesem Ansatz wird eine gemeinsame Masterszene gesucht, jedoch dient diese lediglich als gemeinsame geometrische Referenz und nicht der Erzeugung der D-InSAR-Aufnahmen. Es ist notwendig, eine solche Masterszene auszuwählen, da sämtliche Interferogramme im Bildstapel für die weitere Prozessierung auf dem gleichen Koordinatengitter vorliegen müssen. Somit entsprechen Bildpunkte mit den gleichen Koordinaten in allen Bildern dem gleichen Realweltobjekt. Laut CASU (2009) ist es für die Koregistrierung hilfreich, eine SAR-Aufnahme als Master zu verwenden, die möglichst im Schwerpunkt liegt, wenn die SAR-Szenen in der Basislinienebene auftragen werden (vgl. Abbildung 2.3).

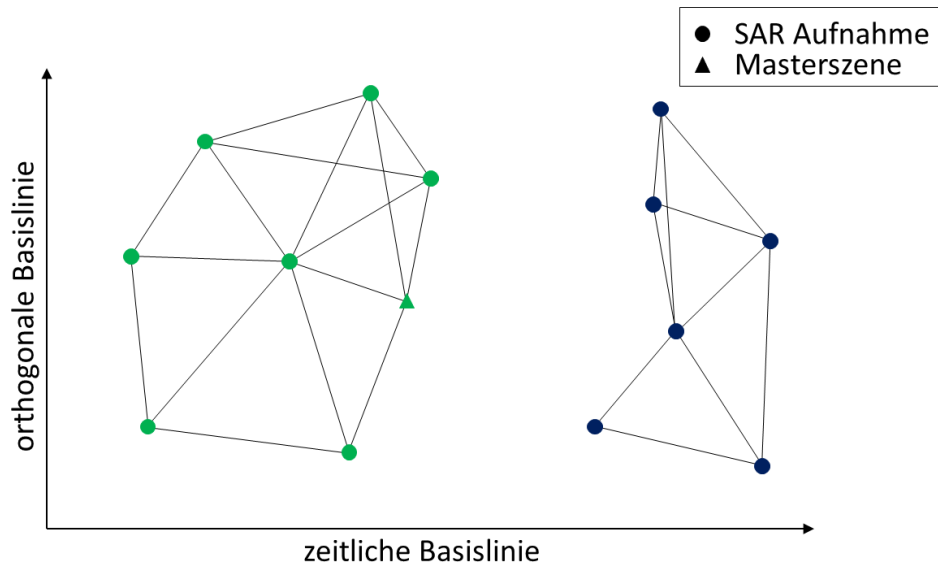


Abbildung 2.3: Basisliniennetzwerk eines Bildstapels, wie er für die SBAS-Methode verwendet wird.

Es wird erneut angenommen, dass N SAR-Szenen vorliegen und bezüglich einer Referenzszene koregistriert sind. Die SAR-Bilder werden als Punkte P_α in der $T \times B_\perp$ -Ebene aufgetragen. Bei der SBAS-Methode kann jede Aufnahme als Masterszene verwendet werden, sodass ein D-InSAR-Bild als Kante zwischen zwei beliebigen Punkten P_α und P_β dargestellt werden kann. Für diese Datenpaare lässt sich eine relative zeitliche Basislinie

$$\Delta t_{\alpha,\beta} = \Delta t_\beta - \Delta t_\alpha = (t_\beta - t_{Master}) - (t_\alpha - t_{Master}) = t_\beta - t_\alpha \quad (2.7)$$

und eine relative orthogonale Basislinie

$$\Delta b_{\perp|\alpha,\beta} = b_{\perp|\beta} - b_{\perp|\alpha} \quad (2.8)$$

bestimmen. Zur Auswahl der Datenpaare, aus denen tatsächlich Interferogramme erzeugt werden sollen, werden Schwellwerte für die maximal erlaubte relative zeitliche Δt_{max} und orthogonale Basislinie $\Delta b_{\perp|max}$ sowie die Doppler-Frequenz-Verschiebung $\Delta f_{c|max}$ festgelegt. Damit soll sichergestellt werden, dass der Dekorrelationseffekt klein bleibt. Dies kann zur Folge haben, dass mehr als ein Subset in der Basislinienebene entsteht (vgl. Abbildung 2.3). Das Auftreten solcher Subsets ist jedoch in der SBAS-Methode vorgesehen (BERARDINO ET AL., 2002; CASU, 2009). Des Weiteren kann die Festlegung der Schwellwerte dazu führen,

dass einzelne SAR-Aufnahmen nicht verwendet werden. Eine weitere Forderung der SBAS-Methode ist, dass die bereits in Abschnitt 2.4 beschriebene Multilook Operation auf den Daten ausgeführt werden muss.

Wird das eben beschriebene Auswahlverfahren verfolgt, entsteht erneut ein Graph in der Basislinienebene, wie Abbildung 2.3 zu entnehmen ist.

In dieser Abbildung ist deutlich zu erkennen, dass die entstehenden relativen Basislinien kleiner sind als die meisten Basislinien im Single-Master-Stapel (vgl. Abbildung 2.2). Dies entspricht der Idee der SBAS-Methode. Außerdem wird offensichtlich, dass sich bei einer gegebenen Anzahl von N' SAR-Aufnahmen keine vordefinierte Anzahl von entstehenden D-InSAR-Bildern angeben lässt. Die Anzahl M' hängt hauptsächlich von der Verteilung der Daten in der $T \times B_{\perp}$ -Ebene ab. Allerdings lässt sich festhalten, dass im Rahmen der SBAS-Methode mehr Interferogramme gebildet werden können als bei einem Single-Master-Stapel und dass das Verhältnis $N' - 1 \leq M'$ gilt.

Bildstapel beim EMCF-Algorithmus:

Der EMCF-Algorithmus fordert neben den Bedingungen, die die SBAS-Methode an die Struktur des D-InSAR-Bildstapels stellt, dass die Punkte P_{α} in der $T \times B_{\perp}$ -Ebene eindeutig vermascht sind.

PEPE UND LANARI (2006) schlagen zur Erzeugung des Datenstapels folgendes Vorgehen vor. Zuerst wird eine Masterszene als geometrische Referenz gesucht, analog zum Vorgehen der SBAS-Methode. Auf diese Weise können die SAR-Daten wieder als Punkte P_{α} in der Basislinienebene dargestellt werden. Anschließend wird auf diesen Punkten eine Delaunay-Triangulation durchgeführt (vgl. Abbildung 2.4).

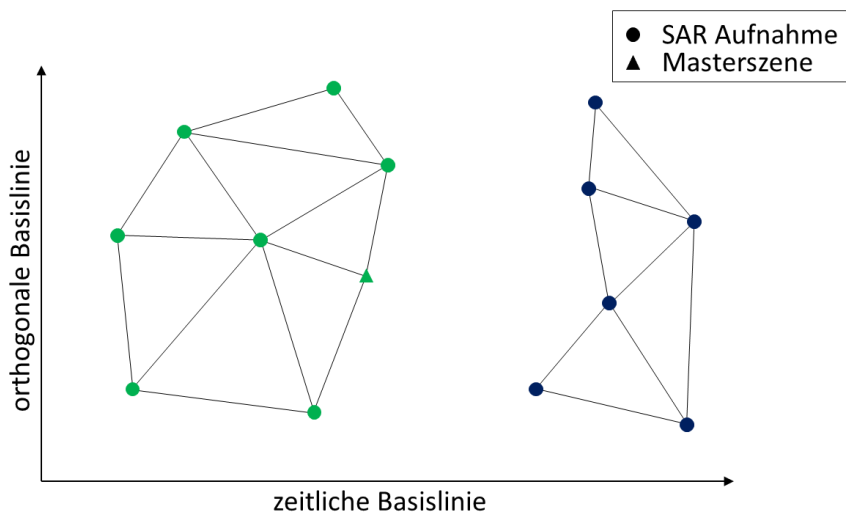


Abbildung 2.4: Basisliniennetzwerk eines Bildstapels, wie er für den EMCF-Algorithmus verwendet wird.

Damit dies funktioniert und zu brauchbaren Ergebnissen führt, müssen die Achsen jedoch die gleiche Einheit haben und über einen ähnlichen Wertebereich verfügen. Um diese Bedingung zu erfüllen, ist im einfachsten Fall ein Skalierungsfaktor einzuführen. Hier bietet es sich an, die zeitliche Achse mit dem Faktor $\Delta t_{max}/\Delta b_{\perp|max}$ umzuskalieren, sodass beide Achsen die Einheit Meter haben. Die Delaunay-Triangulation stellt keine gesonderte Bedingung an die Kantenlängen, sondern folgt nur ihrer Bedingung, den kleinsten Winkel in jedem Dreieck zu maximieren (BERN UND EPPSTEIN, 1992). Aus diesem Grund kann nicht garantiert werden, dass nur SAR-Aufnahmen verbunden werden, deren relative Basislängen und Doppler-Frequenzverschiebung die festgelegten Schwellwerte unterschreiten. Deshalb müssen in einem Nachbearbeitungsschritt alle Kombinationen entfernt werden, die das Schwellwertkriterium verletzen. Sobald eine Kanten entfernt wird, müssen die angrenzenden Dreiecke gelöscht werden (für Kanten der konvexen Hülle gibt es nur ein angrenzendes Dreieck). Es muss jedoch beachtet werden, dass das Löschen eines Dreiecks nicht automatisch

das Löschen aller seiner Kanten zur Folge hat. Eine Kante bleibt solange bestehen, wie sie die Schwellwertkriterien erfüllt und sie noch Teil eines weiteren Dreiecks ist.

Das Löschen von Kanten kann weiterhin dazu führen, dass mehr als eine Dreiecksvermaschung entsteht und sogar einzelne SAR-Szenen nicht verwendet werden. Damit ist der so entstandene Bildstapel konform zu den Forderungen des SBAS-Algorithmus, da er lediglich eine Teilmenge aller möglichen Interferogramme bildet, die die festgelegten Schwellwerte unterschreiten (vgl. Abbildung 2.3 und 2.4).

Die Graphen bzw. Netzwerke in der $T \times B_{\perp}$ -Ebene, die in diesem Abschnitt vorgestellt wurden, werden im Laufe der Arbeit auch als *Basisliniennetzwerke* bezeichnet.

2.5.2 Beschreibung der Datengrundlage

Im Folgenden soll nun die Datengrundlage für diese Arbeit beschrieben und weiterhin gezeigt werden, wie sich die Struktur dieser Daten in die drei beschriebenen Arten von Datenstapeln eingliedern lässt und welche Möglichkeiten sich dadurch für die weitere Verarbeitung eröffnen.

Für diese Arbeit steht ein Datenstapel von 191 differentiellen Radarinterferogrammen zur Verfügung, die von DI Dr. Karlheinz Gutjahr vom Institut für Informations- und Kommunikationstechnologien (DIGITAL) der Forschungsgesellschaft mbH Joanneum Research in Graz, Österreich, mit dem dort entwickelten Softwarepaket *Remote Sensing software package Graz* (RSG) erstellt wurden. Als Grundlage dafür dienten 71 SAR-Bilder der beiden *European Remote sensing Satellite* Missionen (ERS-1 und ERS-2) der europäischen Weltraumorganisation (engl. *European Space Agency*, ESA), die zwischen dem 09.05.1992 und dem 12.12.2000 aufgenommen wurden.

Die verwendeten SAR-Szenen zeigen das Gebiet der Niederrheinischen Bucht, wie es in der Abbildung 2.5 dargestellt wird. Der rote Kasten markiert beispielhaft die Lage der SAR-Aufnahme vom 01.04.1996. Deutlich zu erkennen sind die drei Tagebaue, die das Gebiet prägen. Im Norden der Tagebau Garzweiler und im Süden die Tagebaue Hambach und Inden (von Ost nach West). Ansonsten spiegelt die Abbildung 2.5 den hohen Anteil an vegetativen Flächen wieder, seien es nun Agrar- oder Forstflächen, die zu einem hohem Anteil von dekorierten Bereichen in den differentiellen Interferogrammen führen.

Für die Wahl der Referenzszene auf die alle Aufnahmen koregistriert werden, wird die sogenannte *stack coherence* γ^m verwendet (KAMPES, 2005, S. 6–7):

$$\gamma^m = \frac{1}{N'} \sum_{\alpha=1}^{N'} g(b_{\perp|\alpha}, b_{\perp|krit}) \times g(\Delta t_{\alpha}, \Delta t_{krit}) \times g(\Delta f_{c|\alpha}, \Delta f_{c|krit}) \quad (2.9)$$

wobei

$$g(x, c) = \begin{cases} 1 - |x|/c & , \text{ wenn } |x| < c \\ 0 & , \text{ sonst} \end{cases} \quad (2.10)$$

Die Variablen $b_{\perp|krit}$, Δt_{krit} und $\Delta f_{c|krit}$ stehen dabei für die sogenannten kritischen Basislinien, für die totale Dekorrelation erwartet wird. Nach HOOPER (2006, S. 36) sind für ERS-Daten $b_{\perp|krit} = 1100m$, $\Delta t_{krit} = 5$ Jahre und $\Delta f_{c|krit} = 1380Hz$ typische Werte.

Wie der Formel 2.9 zu entnehmen ist, wird der Wert γ^m groß, wenn die Masterszene so gewählt wird, dass die Basislinien und die Doppler-Frequenz-Verschiebung klein werden. Diese Methode zur Festlegung der Masterszene hat KAMPES (2005) eigentlich für die Anwendung des STUN-Algorithmus vorgesehen, sie lässt sich jedoch auch zur Wahl der Referenzszene für Small Baseline-Techniken nutzen. Auf diese Weise wurde für den vorliegenden Stapel die Referenzszene auf die Aufnahme vom 18.03.1997 festgelegt.

Zur Elimination des Phasenanteils der Topographie wurde ein Digitales Höhenmodell (DHM) verwendet, welches aus den Daten der *Shuttle Radar Topographic Mission* (SRTM) der US-amerikanischen Luft- und Raumfahrtbehörde (engl. *National Aeronautics and Space Administration*, NASA) erzeugt wurde.

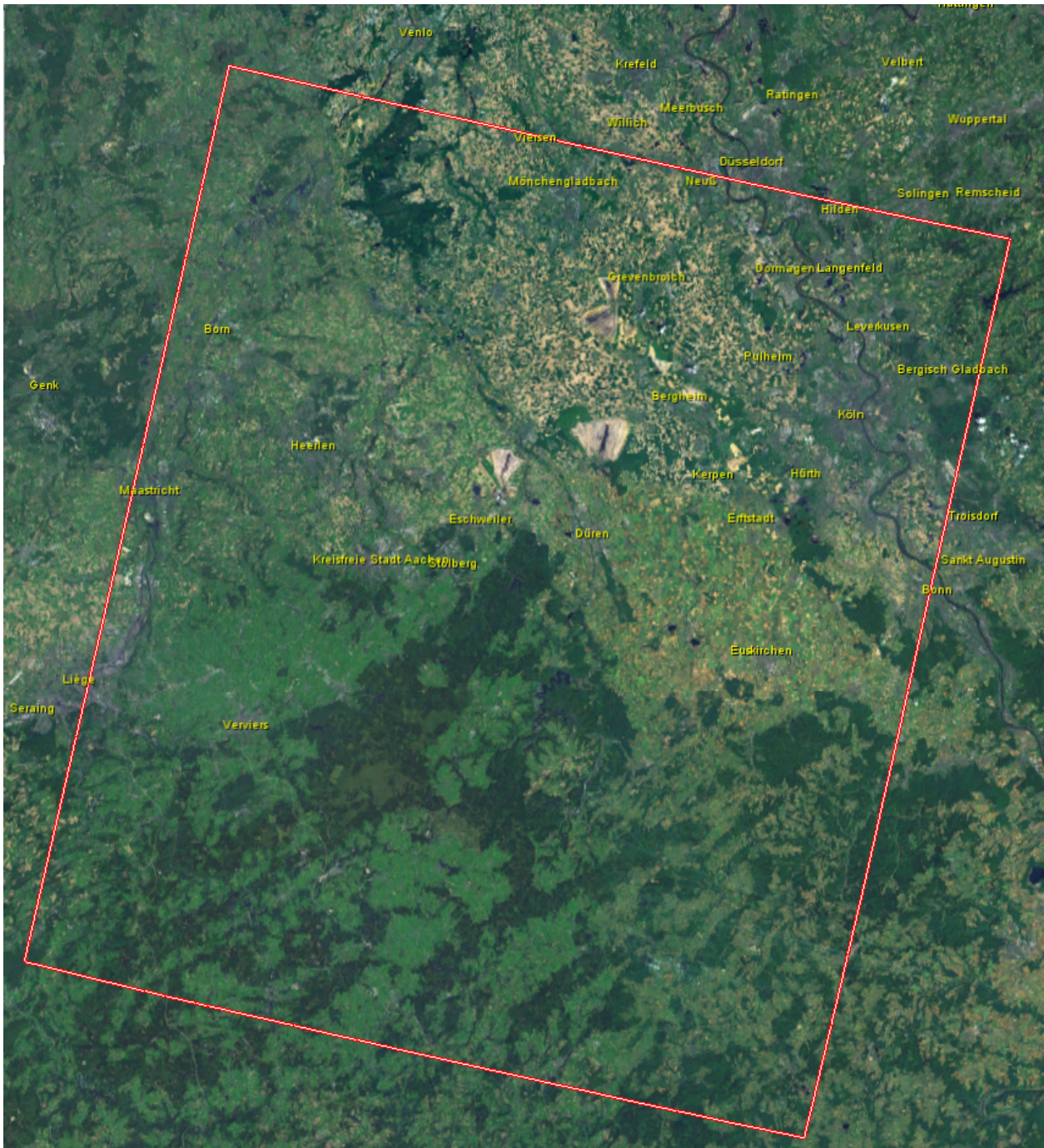


Abbildung 2.5: Übersicht des Aufnahmegebiets der Niederrheinischen Bucht: Das rote Rechteck markiert die Lage der SAR-Szene vom 01. April 1996. Das Bild wurde mit der NEST Toolbox Version 5.1 erzeugt (EUROPEAN SPACE AGENCY (ESA), 2015).

Des Weiteren liegen die erzeugten D-InSAR-Bilder als sogenannte *80-Look*-Daten vor. Das bedeutet, auf den Daten wurde eine Multilook Operation ausgeführt, um das Rauschen in den Interferogrammen zu vermindern. Eine *Single Look* ERS-Aufnahme hat eine Auflösung von etwa $4m$ in Azimuth- und $20m$ in Rangerichtung.

Daher wird für ERS-Daten in aller Regel ein Verhältnis von 1 (Rangerichtung) zu 5 (Azimuthrichtung) bei der Multilook-Bildung gewählt, sodass annähernd quadratische Pixel entstehen. Bei 80-Look werden 4×20 Pixel (Looks) verwendet, was zu einer Auflösung von etwa $80m \times 80m$ führt.

Neben den differentiellen Interferogrammen sind auch die zugehörigen Intensitäts- und Kohärenzbilder gegeben. Außerdem liegt eine Textdatei vor, die den Datenstapel beschreibt und in der zeilenweise für jedes Interferogramm die in Tabelle 2.1 angegebenen Informationen gespeichert sind.

Tabelle 2.1: Beschreibung des gegebenen Datenstapels beispielhaft für die ersten zehn Interferogramme.

ID (Slave)	ID (Master)	$\Delta t_{\alpha,\beta}$ [d]	$\Delta b_{\perp \alpha,\beta}$ [m]	$\Delta f_{c \alpha,\beta}$ [Hz]	Aufnahmedatum (Slave)	Aufnahmedatum (Master)
2	1	35	8.590	44.420	13-JUN-1992	9-MAY-1992
3	1	70	367.698	12.338	18-JUL-1992	9-MAY-1992
3	2	35	359.095	32.082	18-JUL-1992	13-JUN-1992
4	1	140	11.233	0.485	26-SEP-1992	9-MAY-1992
4	2	105	2.597	44.905	26-SEP-1992	13-JUN-1992
4	3	70	-356.154	12.823	26-SEP-1992	18-JUL-1992
6	1	210	-383.310	1.418	5-DEC-1992	9-MAY-1992
7	3	175	27.150	7.038	9-JAN-1993	18-JUL-1992
7	4	105	382.988	19.861	9-JAN-1993	26-SEP-1992
8	5	140	449.219	15.590	20-MAR-1993	31-OCT-1992
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Wie der Tabelle 2.1 zu entnehmen ist, wurde bei der Interferogrammbildung als Masterszene stets die SAR-Szene benutzt die chronologisch vor der Slaveszene liegt. Dadurch entstehen durchweg positive, relative zeitliche Basislinien. Zur Untersuchung der Struktur des Datenstapels, werden die Koordinaten der SAR-Bilder in der $T \times B_{\perp}$ -Ebene benötigt. Diese lassen sich mit der Kenntnis der Referenzszene m , deren Koordinaten mit $P_m = (0, 0)$ definiert sind, und den relativen Basislinien aus Tabelle 2.1 über die Lösung eines überbestimmten, linearen Gleichungssystems ermitteln (vgl. Gleichungen (2.7) und (2.8)).

Werden die SAR-Szenen nun in die Basislinienebene übertragen und die Interferogramme entsprechend nach Tabelle 2.1 eingetragen, so ergibt sich für die Struktur des D-InSAR-Bildstapels die Abbildung 2.6.

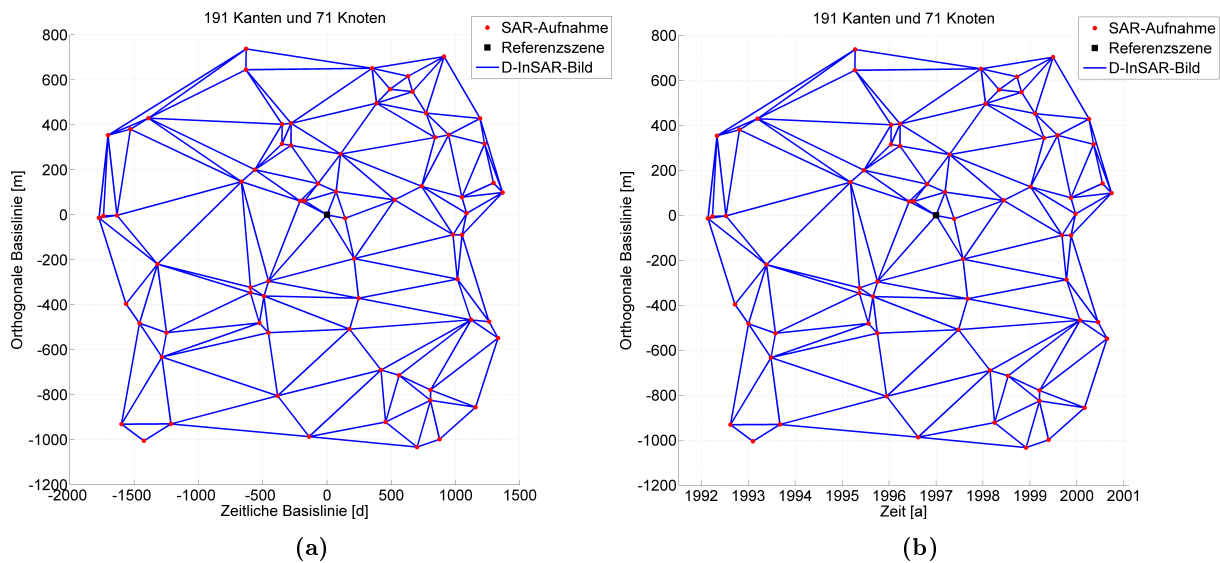


Abbildung 2.6: Struktur des gegebenen Datenstapels in der Basislinienebene. Der D-InSAR-Bildstapel ist mit (a) relativem Zeitbezug zur Referenzszene und mit (b) absolutem Zeitbezug, also dem Aufnahmedatum, dargestellt.

Die beiden Abbildungen 2.6a und 2.6b unterscheiden sich lediglich im gewählten Zeitbezug. Während für Abbildung 2.6a die relative zeitliche Basislinie verwendet wird, womit diese mit den Ausführungen in Abschnitt 2.5.1 übereinstimmt, wird in Abbildung 2.6b für die Zeitachse das Aufnahmedatum benutzt, damit die Verteilung der Daten im gegebenen Aufnahmezeitraum (09.05.1992 bis 12.12.2000) deutlicher wird. In den Abbildungen ist zu erkennen, dass der vorliegende Bildstapel für eine Auswertung mit der SBAS-Methode erzeugt wurde, da kein Sternnetzwerk vorliegt. Weiterhin bilden die Interferogramme eine Triangulation der SAR-Punkte in der $T \times B_{\perp}$ -Ebene, womit eine Mehrdeutigkeitslösung mit dem EMCF-Algorithmus ermöglicht wird. Es liegen keine weiteren Informationen zur Entstehung der Triangulation vor, was für die weitere Auswertung auch nicht von Bedeutung ist.

Zur vollständigen Beschreibung des Datenstapels werden abschließend noch die Wertebereiche der Basislinien und der Doppler-Frequenz-Verschiebung genannt. Während die relative orthogonale Basislinie zwischen $-468.608m$ und $449.219m$ liegt, nimmt die relative zeitliche Basislinie Werte zwischen einem und 1074 Tagen an. Die Doppler-Frequenz-Verschiebung beträgt zwischen $0,485Hz$ und $1003.883Hz$.

3. Prozessierung von D-InSAR-Bildstapel

Im vorangegangenen Kapitel wurde die Theorie vermittelt, die zum Verständnis der vorliegenden Datengrundlage dieser Arbeit und dessen Struktur nötig ist. Für die Entwicklung einer Datenstruktur sind außerdem Kenntnisse über die Arbeitsschritte erforderlich, die auf einen D-InSAR-Bildstapel angewendet werden können. Gerade diese Anwendungen definieren, welche Informationen in der Datenstruktur gespeichert und welche Anfragen an die Datenstruktur unterstützt werden müssen.

Es wurde bereits erwähnt, dass der Schritt des Phase Unwrappings (dt. *Phasenabwicklung*) ein wichtiger und genauigkeitsbestimmender Schritt in der Datenprozessierung von SAR-Interferogrammen ist. Daher soll im Folgenden die Problemstellung des Phase Unwrappings sowie die in dieser Arbeit gewählte mathematische Notation vorgestellt werden (Abschnitt 3.1). Im Anschluss daran erfolgt die Darstellung der Phasenabwicklung als *Minimum Cost Flow* (MCF)-Problem, die eine Klasse von Methoden zur Lösung der Phasenmehrdeutigkeiten motiviert (vgl. Abschnitt 3.2). In den Abschnitten 3.3 und 3.4 werden zwei spezielle Methoden vorgestellt, die für das Phase Unwrapping von Bildstapeln entwickelt wurden und somit letztlich Anforderungen an die zu entwickelnde Datenstruktur stellen.

Nach dem Lösen der Phasenmehrdeutigkeiten ist die SBAS-Technik der nächste Prozessierungsschritt, der auf den gegebenen Bildstapel angewendet werden kann. Mit ihr ist es möglich, aus einem Stapel von abgewickelten D-InSAR-Aufnahmen die gewünschte Deformationszeitreihe zu erzeugen. Deshalb soll die SBAS-Methode in Abschnitt 3.5 vorgestellt werden.

3.1 Allgemeine Grundlagen des Phase Unwrappings

Als Phase Unwrapping wird der Prozessierungsschritt bezeichnet, der aus den mehrdeutigen Phasenwerten der (differentiellen) SAR-Interferogramme (vgl. Abschnitt 2.2 und 2.3) absolute Phasen berechnet. Dabei handelt es sich eigentlich um ein nicht lösbares Problem, da im gesuchten abgewickelten Phasenfeld mehr Information enthalten ist als in den mehrdeutigen Interferogrammen. Die gängigste Annahme, die getroffen wird, um dennoch eine Lösung zu erhalten, basiert auf dem *Nyquist-Shannon-Abtasttheorem*. Es wird angenommen, dass die räumliche Abtastung in alle Richtungen hoch genug ist, sodass kein Aliasing-Effekt auftritt. Somit kann angenommen werden, dass sich die wahren abgewickelten Phasen zweier Pixel um maximal einen halben Zyklus unterscheiden (CHEN, 2001, S. 8). Diese Annahme wird in aller Regel an einigen Stellen verletzt. Bei D-InSAR-Bildern ist dies der Fall, wenn die zu detektierende Höhenänderung zwischen zwei Pixeln zu groß wird. Welcher Gradient von Höhenänderungen aufgelöst werden kann, hängt von diversen Kriterien, wie beispielsweise der Wellenlänge λ des Radars, der Bodenauflösung und -neigung ab (SCHÄFER, 2012, S. 105 ff). Auch ein hohes Phasenrauschen durch niedrige Kohärenz (vgl. Abschnitt 2.4) kann dazu führen, dass der Phasengradient zwischen zwei Pixeln nicht im definierten Wertebereich von $[-\pi, \pi)$ liegt (CASU, 2009, S. 75). Stellen, an denen die Annahme verletzt ist, werden in der Literatur als Residuen oder Diskontinuitäten bezeichnet.

Wäre das Abtasttheorem überall in einem Bild erfüllt, so ließe sich aus dem mehrdeutigen Phasenbild durch Aufintegration der Phasendifferenzen benachbarter Pixel das absolute Phasenbild bestimmen. Doch durch die auftretenden Residuen ist die Integration pfadabhängig und würde bei nicht Beachtung der Residuen zu absoluten Phasenfehlern führen. Diese nehmen Vielfache von ganzen Phasenzyklen an und pflanzen sich entlang des Integrationspfads fort. Daher gilt die Phasenabwicklung auch als genauigkeitsbestimmender Prozessierungsschritt (MOLL, 2007, S. 41 ff).

Für ein differentielles Interferogramm, das aus zwei SAR-Szenen mit den Aufnahmezeitpunkten t_α und t_β

erzeugt wurde, lässt sich der absolute, abgewickelte Phasenwert (engl. *unwrapped phase*) $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ an einem Pixel $x(i,j)$ über

$$\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}} = \psi_{x(i,j)}^{\Delta t_{\alpha,\beta}} + 2\pi k_{x(i,j)}^{\Delta t_{\alpha,\beta}} \quad (3.1)$$

ins Verhältnis zur gegebenen, nicht-abgewickelten Phase (engl. *wrapped phase*) $\psi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ setzen. Dabei ist $k_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ der sogenannten Phasenmehrdeutigkeitsfaktor und es gilt $k_{x(i,j)}^{\Delta t_{\alpha,\beta}} \in \mathbb{Z}$. Nun versuchen Phase Unwrapping-Methoden nicht direkt die abgewickelte Phase aus den Interferogrammen zu bestimmen, sondern sie schätzen den abgewickelten Phasengradienten aus der Differenz der interferometrischen Phasen zweier Pixel $x(i,j)$ und $x(i',j')$. Der Zusammenhang der Gradienten lässt sich ausgehend von Formel (3.1) über

$$\begin{aligned} \phi_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}} &= \phi_{x(i',j')}^{\Delta t_{\alpha,\beta}} - \phi_{x(i,j)}^{\Delta t_{\alpha,\beta}} \\ &= \langle (\psi_{x(i',j')}^{\Delta t_{\alpha,\beta}} + 2\pi k_{x(i',j')}^{\Delta t_{\alpha,\beta}}) - (\psi_{x(i,j)}^{\Delta t_{\alpha,\beta}} + 2\pi k_{x(i,j)}^{\Delta t_{\alpha,\beta}}) \rangle_{-\pi,\pi} \\ &= \langle \psi_{x(i',j')}^{\Delta t_{\alpha,\beta}} - \psi_{x(i,j)}^{\Delta t_{\alpha,\beta}} \rangle_{-\pi,\pi} + 2\pi k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}} \\ &= \psi_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}} + 2\pi k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}} \end{aligned} \quad (3.2)$$

darstellen, wobei $\langle \cdot \rangle_{-\pi,\pi}$ den 2π -Modulo-Operator repräsentiert, der einen Phasenwert in den mehrdeutigen Wertebereich von $[-\pi, \pi)$ überführt. Wie aus Formel (3.2) zu entnehmen ist, entspricht der gewrappte Phasengradient $\psi_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$ nicht der Differenz der mehrdeutigen Phasen $\psi_{x(i',j')}^{\Delta t_{\alpha,\beta}}$ und $\psi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$. Aus diesem Grund kann sich auch der Mehrdeutigkeitsfaktor $k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$ um ± 1 von der Differenz der Faktoren aus Formel (3.1) unterscheiden.

Der Mehrdeutigkeitsfaktor $k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$ ist die mathematische Darstellung der vorher angesprochenen Diskontinuität. Das Ziel eines Phase Unwrapping Algorithmus ist es, diese Diskontinuitäten zu finden und die Gradienten mit dem richtigen Faktor zu korrigieren (CHEN, 2001, S. 11).

3.2 Darstellung des Phase Unwrappings als MCF-Problem

Grundsätzlich lassen sich die meisten Phase Unwrapping-Algorithmen in drei Klassen einteilen, die *Branch-Cut*-Methoden (auch *Residue-Cut*- oder *Path-Following*-Methoden genannt), die Kleinste-Quadrate-Methoden und die *Network Flow*-Methoden. Eine Einführung in die Grundkonzepte der einzelnen Klassen ist in ROSEN ET AL. (2000), CHEN (2001) und RICHARDS (2006) zu finden. Die Beschreibung in ROSEN ET AL. (2000) ist zwar sehr kurz, wird dafür durch eine umfangreiche Bibliographie ergänzt, die weitergehende Recherchen erleichtert.

In diesem Abschnitt soll allerdings nur auf die ursprüngliche Darstellung des Phase Unwrappings als *Minimum Cost Flow* (MCF)-Problem von COSTANTINI (1997) eingegangen werden oder vielmehr auf dessen Spezialisierung für dünnbesetzte Interferogramme (COSTANTINI UND ROSEN, 1999). Auf die allgemeine Darstellung der mathematischen Grundlagen von MCF-Problemen und der ganzzahligen linearen Optimierung wird hier verzichtet und auf PAPADIMITRIOU UND STEIGLITZ (1998) verwiesen. Einen guten Einstieg im Hinblick auf die Anwendung der linearen Programmierung für das Phase Unwrapping bietet auch WALZOG (2014).

Die von COSTANTINI UND ROSEN (1999) vorgestellte Phase Unwrapping-Methode ist ein sehr effizientes Verfahren zur Mehrdeutigkeitslösung von dünnbesetzten Interferogrammen, wie sie bei der Auswertung von kohärenten Pixeln oder der Permanent Scatterer Interferometrie auftreten. Durch die Darstellung des Phase Unwrappings als (ganzzahliges) MCF-Problem kann auf eine Vielzahl von leistungsfähigen Algorithmen zurückgegriffen werden.

Ausgegangen wird von der Erzeugung eines Graphen auf einem Interferogramm mittels einer Triangulation (z.B. Delaunay Triangulation), die auf den kohärenten Punkten (oder den PS-Punkten) durchgeführt wird. Dadurch entsteht in der durch die Achsen des Bildes aufgespannten $A_Z \times R_G$ -Ebene (Azimuth/Range-Ebene) ein Graph, bestehend aus M Kanten und N Dreiecken. Die Knoten des Graphen entsprechen dabei den gesuchten absoluten Phasenwerten und die Kanten den abgewickelten Phasengradienten (vgl. Abbildung 3.1).

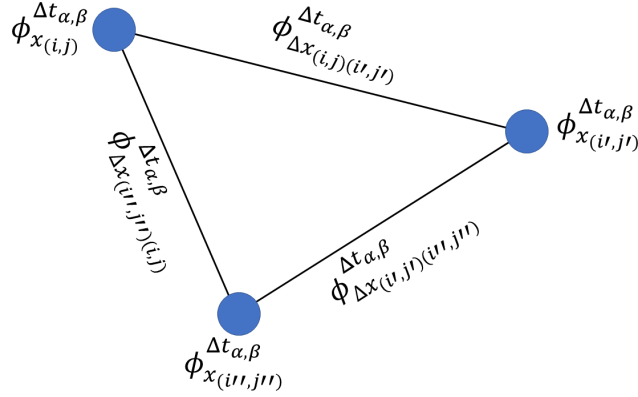


Abbildung 3.1: Dreieck einer räumlichen Triangulation auf den kohärenten Pixeln.

Die Mehrdeutigkeitsfaktoren $k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$ aus Gleichung (3.2) stellen die Variablen dar, bezüglich derer der Kostenfluss optimiert werden soll. Als Zielfunktion wird die Minimierung der gewichteten Summe der Residuen gewählt:

$$\sum_{\forall \Delta x(i,j)(i',j') \in M} w_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}} \left| k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}} \right| \dots \min. \quad (3.3)$$

In Gleichung (3.3) stehen die $w_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$ für die Gewichte, die bei geeigneter Wahl den Fluss über die Kanten steuern können. Ein hohes Gewicht hat zur Folge, dass eine Korrektur des Phasengradienten an der zugehörigen Kante unwahrscheinlicher wird, wohingegen ein niedriges Gewicht die Wahrscheinlichkeit für einen Mehrdeutigkeitsfaktor ungleich Null erhöht.

Zur Lösung des MCF-Problems muss allerdings noch eine Restriktion eingeführt werden, die dafür sorgt, dass die aus dem Minimierungsproblem rekonstruierte, abgewickelte Phase nicht vom Integrationspfad abhängt. Diese Restriktion muss in allen Dreiecken der Triangulation erfüllt sein und ist gegeben durch

$$\phi_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}} + \phi_{\Delta x(i',j')(i'',j'')}^{\Delta t_{\alpha,\beta}} + \phi_{\Delta x(i'',j'')(i,j)}^{\Delta t_{\alpha,\beta}} = 0. \quad (3.4)$$

Setzt man den Zusammenhang zwischen den absoluten und mehrdeutige Phasengradienten aus Formel (3.2) in Gleichung (3.4) ein, so lässt sich für jedes der N Dreiecke mit

$$k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}} + k_{\Delta x(i',j')(i'',j'')}^{\Delta t_{\alpha,\beta}} + k_{\Delta x(i'',j'')(i,j)}^{\Delta t_{\alpha,\beta}} = - \left[\frac{\psi_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}} + \psi_{\Delta x(i',j')(i'',j'')}^{\Delta t_{\alpha,\beta}} + \psi_{\Delta x(i'',j'')(i,j)}^{\Delta t_{\alpha,\beta}}}{2\pi} \right] \quad (3.5)$$

eine Restriktionsgleichung aufstellen, die als Unbekannte die Mehrdeutigkeitsfaktoren enthält. Dabei steht $[\cdot]$ für den Rundungsoperator zum nächsten Integer, der numerische Ungenauigkeiten ausgleicht und sicherstellt, dass die geschätzten Variablen tatsächlich Integerwerte sind.

Da die Mehrdeutigkeitsfaktoren $k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$ sowohl positiv, als auch negativ sein können (vgl. Gleichung (3.2)) liegt ein nicht-lineares Minimierungsproblem mit Integer-Variablen vor. Um dieses in ein gültiges

MCF-Problem zu überführen, sieht COSTANTINI (1997) vor, die Variablen in positive und negative Faktoren über

$$\begin{aligned} k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta,+}} &= \max\left(0, k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}\right) \quad \text{und} \\ k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta,-}} &= -\min\left(0, k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}\right) \end{aligned} \quad (3.6)$$

aufzuspalten. Werden in Gleichung (3.3) und (3.4) nun die Variablen durch

$$k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}} = \left(k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta,+}} - k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta,-}}\right) \quad (3.7)$$

ersetzt, so ergibt sich eine MCF-Formulierung für das Phase Unwrapping, das sich eindeutig als ganzzahliges lineares Programm lösen lässt.

3.3 Der Extended Minimum Cost Flow Algorithmus

Im vorangegangenen Abschnitt wurde gezeigt, wie sich das Phase Unwrapping für ein einzelnes dünnbesetztes Interferogramm als MCF-Problem in der $A_Z \times R_G$ -Ebene aufstellen lässt. Nun soll gezeigt werden, wie sich dieser Ansatz auf einen Bildstapel von D-InSAR-Szenen übertragen lässt.

Die zusätzliche Nutzung der zeitlichen Zusammenhänge zwischen den Interferogrammen bei der Phasenabwicklung erhöht die Zuverlässigkeit des Ergebnisses, also die Wahrscheinlichkeit, dass Diskontinuitäten korrekt detektiert und korrigiert werden. Dazu soll hier der *Extended Minimum Cost Flow* (EMCF)-Algorithmus von PEPE UND LANARI (2006) vorgestellt werden, der die in dieser Arbeit entwickelte Datenstruktur (vgl. Kapitel 4) maßgeblich beeinflusst hat.

Bereits in Abschnitt 2.5.1 wurde gezeigt, dass sich ein Bildstapel von (differentiellen) SAR-Interferogrammen als Graph in einer Ebene ($T \times B_{\perp}$ -Ebene) darstellen lässt, die von den Basislinien bezüglich einer Referenzszenen aufgespannt wird. Außerdem wurde erläutert, dass dieser Graph bei einer geeigneten Wahl von Datenpaaren zur Interferogrammerzeugung einer Triangulation entspricht. Diese Triangulation enthält Informationen zu den zeitlichen und räumlichen Zusammenhängen zwischen den Interferogrammen, die vom EMCF-Algorithmus genutzt werden, um eine zuverlässigere Phasenabwicklung zu gewährleisten. Zudem wird eine weitere Triangulation in der Bildebene benötigt, die von der Azimuth- und Rangerichtung aufgespannt wird.

Die Punkte, auf denen die Dreiecksvermaschung ausgeführt wird, werden anhand ihrer Kohärenzwerte gewählt. Dafür wird in einem ersten Schritt ein Kohärenzschwellwert θ und ein prozentualer Anteil p festgelegt. Nun wird für jede Bildkoordinate $x_{(i,j)}$ die Bildsäule untersucht, die von den Pixeln entlang des Stapels gebildet wird, und überprüft, ob in dieser Säule der Schwellwert θ von mindestens $p\%$ der Pixel überschritten wird. Die Bildkoordinaten $x_{(i,j)}$, welche die Kriterien erfüllen, werden beispielsweise über eine Delaunay-Triangulation vermascht. Jede der Koordinaten entspricht jeweils einem Phasenwert in einem der M' Interferogrammen und für jede der Kanten der Triangulation kann jeweils ein gewrappter Phasengradient pro Bild berechnet werden. Somit werden eigentlich M' Triangulationen erzeugt, die allerdings für alle D-InSAR-Bilder identisch sind.

Um die Informationen aus der Dreiecksvermaschung in der $T \times B_{\perp}$ -Ebene für das Phase Unwrapping zu nutzen, muss diese noch mit den Phasengradienten in der $A_Z \times R_G$ -Ebene in Verbindung gebracht werden. Abbildung 3.2 verdeutlicht diesen Zusammenhang anhand eines Minimalbeispiels. Auf der linken Seite ist ein Bildstapel mit drei D-InSAR-Interferogrammen zu sehen, in denen für drei lagegleiche, kohärente Punkte eine Triangulation in der $A_Z \times R_G$ -Ebene durchgeführt wurde. Rechts ist das zum Bildstapel korrespondierende Netzwerk in der $T \times B_{\perp}$ -Ebene dargestellt. In Abschnitt 2.5.1 wurde diskutiert, dass sich

die Knoten in diesem Netzwerk als SAR-Bilder zu einem Zeitpunkt t_α mit $\alpha = 1 \dots N'$ auffassen lassen, wobei N' die Anzahl der SAR-Aufnahmen angibt. Die Kanten repräsentieren die Interferogramme, die aus den SAR-Szenen erzeugt wurden, welche durch die Kante verbunden werden. Wird nun im Bildstapel eine Menge von lageidentischen Phasengradienten betrachtet (hier in rot dargestellt), so lässt sich jeder dieser Gradienten derjenigen Kante im rechten System zuordnen, die dem D-InSAR-Bild entspricht, aus dem er abgegriffen wurde. Somit steht mit dem Basisliniennetz für jede Menge lagegleicher Kanten ein Graph zur Verfügung, in dem die zeitlichen Zusammenhänge für das Phase Unwrapping genutzt werden können.

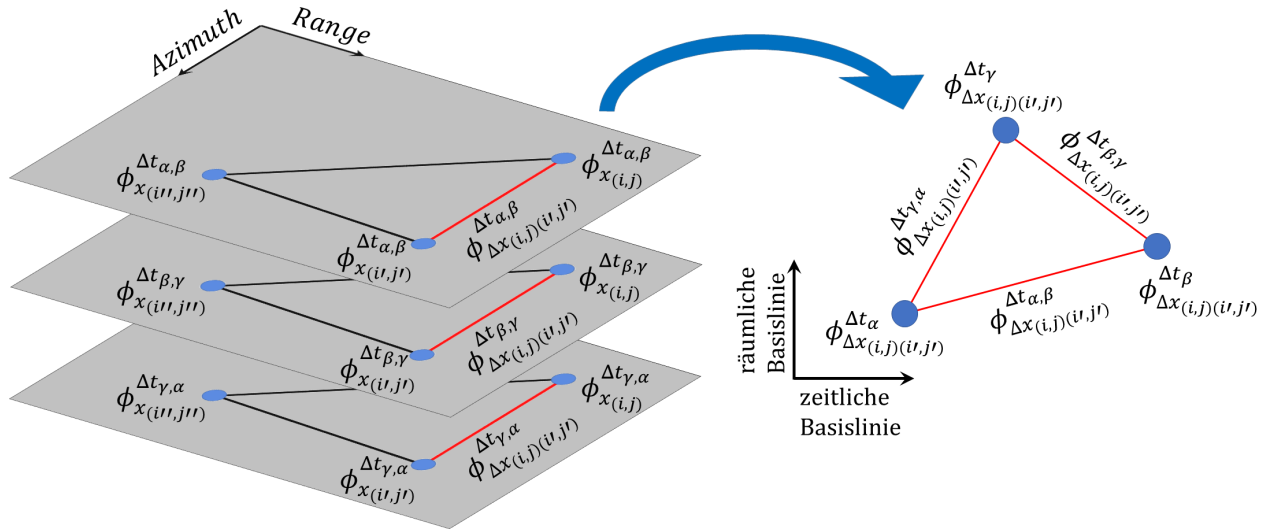


Abbildung 3.2: Zusammenhang zwischen den Phasengradienten in den räumlichen Triangulationen und dem Bildstapel-Netzwerk in der Basislinienebene.

Genau diesen Umstand machen sich PEPE UND LANARI (2006) zu Nutze. Dazu unterteilen sie den Algorithmus in zwei Schritte. Im sogenannten *Temporal Unwrapping* (dt. *Zeitliche Abwicklung*) wird für jede Menge von lagegleichen Kanten ein MCF-Problem im Basisliniennetz gelöst. Die Lösungen für die Phasengradienten aus diesem Schritt bilden dann eine erste Näherungslösung für das *Spatial Unwrapping* (dt. *Räumliche Abwicklung*). Dieses wird für jedes D-InSAR-Bild mit der MCF-Methode (vgl. Abschnitt 3.2) durchgeführt.

3.3.1 Temporal Unwrapping

Angenommen es liegt ein Bildstapel mit M' D-InSAR-Szenen vor, dessen Struktur die Voraussetzungen für den EMCF-Algorithmus erfüllt (vgl. Abschnitt 2.5.1). Sollte das Netz in der $T \times B_\perp$ -Ebene aus mehreren Teilnetzen bestehen, so würde der Algorithmus separat für die mit den Teilnetzen korrespondierenden Interferogramme durchgeführt werden. Daher wird sich hier auf die Beschreibung der EMCF-Methode für einen zusammenhängenden Bildstapel beschränkt.

Die relativen zeitlichen und örtlichen Basislinien, die den D-InSAR-Bildern zugeordnet werden können, werden in jeweils einem Vektor $\Delta \mathbf{t}$ beziehungsweise $\Delta \mathbf{b}_\perp$ zusammengefasst.

Weiterhin wird angenommen, dass die kohärenten Punkte, wie oben beschrieben, ausgewählt wurden und die Triangulation in der $A_Z \times R_G$ -Ebene durchgeführt wurde. Betrachtet man nun eine Menge von lagegleichen Kanten aus den räumlichen Triangulationen wie in Abbildung 3.2, die von den Pixeln $x_{(i,j)}$ und $x_{(i',j')}$ aufgespannt werden, so lassen sich diese einer Menge von mehrdeutigen Phasengradienten

$$\psi_{\Delta x(i,j)(i',j')}^{\Delta \mathbf{t}} = \left[\psi_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}, \psi_{\Delta x(i,j)(i',j')}^{\Delta t_{\beta,\gamma}}, \psi_{\Delta x(i,j)(i',j')}^{\Delta t_{\gamma,\alpha}}, \dots \right]^T \quad (3.8)$$

zuordnen. Die gesuchten, absoluten Phasengradienten lassen sich durch ein Modell über

$$\begin{aligned} \phi_{\Delta x_{(i,j)(i',j')}}^{\Delta t} &\approx \mathbf{m}(\Delta z_{\Delta x_{(i,j)(i',j')}} , \Delta v_{\Delta x_{(i,j)(i',j')}}) \\ &\approx \frac{4\pi}{\lambda} \cdot \left(\frac{\Delta \mathbf{b}_{\perp}}{r \cdot \sin(\vartheta)} \cdot \Delta z_{\Delta x_{(i,j)(i',j')}} + \Delta \mathbf{t} \cdot \Delta v_{\Delta x_{(i,j)(i',j')}} \right) \end{aligned} \quad (3.9)$$

mit dem topographischen Fehler $\Delta z_{\Delta x_{(i,j)(i',j')}}$ und der Änderung der Deformationsgeschwindigkeit $\Delta v_{\Delta x_{(i,j)(i',j')}}$ entlang der betrachteten Kante zwischen $x_{(i,j)}$ und $x_{(i',j')}$ annähern. Des Weiteren steht λ für die vom Radar genutzte Wellenlänge, r für die Sensor-Ziel-Distanz und ϑ für den lokalen Einfallswinkel. Da sich der absolute und der mehrdeutige Phasengradient nur um ein Vielfaches von 2π unterscheiden (vgl. Formel (3.2)), kann man den abgewickelten Phasengradienten im Zusammenhang mit dem Modell auch wie folgt ausdrücken:

$$\begin{aligned} \phi_{\Delta x_{(i,j)(i',j')}}^{\Delta t} &= \mathbf{m} + \langle \psi_{\Delta x_{(i,j)(i',j')}}^{\Delta t} - \mathbf{m} \rangle_{-\pi, \pi} + 2\pi \mathbf{k}_{\Delta x_{(i,j)(i',j')}}^{\Delta t} \\ &= \chi_{\Delta x_{(i,j)(i',j')}}^{\Delta t} + 2\pi \mathbf{k}_{\Delta x_{(i,j)(i',j')}}^{\Delta t} \end{aligned} \quad (3.10)$$

mit dem Vektor von abgeleiteten Beobachtungen

$$\chi_{\Delta x_{(i,j)(i',j')}}^{\Delta t} = \mathbf{m} + \langle \psi_{x_{(i,j)(i',j')}}^{\Delta t} - \mathbf{m} \rangle_{-\pi, \pi}, \quad (3.11)$$

die den modellierten mit dem gemessenen Phasengradienten in Verbindung bringen. Aus Gründen der Übersichtlichkeit wurde in Formel (3.10) und (3.11) das Modell $\mathbf{m}(\Delta z_{\Delta x_{(i,j)(i',j')}} , \Delta v_{\Delta x_{(i,j)(i',j')}})$ mit \mathbf{m} abgekürzt. Ausgehend hiervon lässt sich das zeitliche Phase Unwrapping als Kostenminimierungsproblem auf dem Basisliniennetzwerk beschreiben. Die Zielfunktion stellt die Minimierung der Gesamtkosten (analog zu Formel (3.3)) über

$$C_{\Delta x_{(i,j)(i',j')}} = \sum_{\forall \Delta t_{\alpha, \beta} \in M'} \left| k_{\Delta x_{(i,j)(i',j')}}^{\Delta t_{\alpha, \beta}} \right| \dots \min \quad (3.12)$$

dar. Diese ist abhängig von den Restriktionen, die besagen, dass die absoluten Phasengradienten in geschlossenen Schleifen verschwinden müssen:

$$\phi_{\Delta x_{(i,j)(i',j')}}^{\Delta t_{\alpha, \beta}} + \phi_{\Delta x_{(i,j)(i',j')}}^{\Delta t_{\beta, \gamma}} + \phi_{\Delta x_{(i,j)(i',j')}}^{\Delta t_{\gamma, \alpha}} = 0. \quad (3.13)$$

Nach einsetzen von Gleichung (3.10) in (3.13) ergeben sich die Restriktionen zu

$$k_{\Delta x_{(i,j)(i',j')}}^{\Delta t_{\alpha, \beta}} + k_{\Delta x_{(i,j)(i',j')}}^{\Delta t_{\beta, \gamma}} + k_{\Delta x_{(i,j)(i',j')}}^{\Delta t_{\gamma, \alpha}} = - \left[\frac{\chi_{\Delta x_{(i,j)(i',j')}}^{\Delta t_{\alpha, \beta}} + \chi_{\Delta x_{(i,j)(i',j')}}^{\Delta t_{\beta, \gamma}} + \chi_{\Delta x_{(i,j)(i',j')}}^{\Delta t_{\gamma, \alpha}}}{2\pi} \right]. \quad (3.14)$$

Da die Variablen $(\Delta z_{\Delta x_{(i,j)(i',j')}} , \Delta v_{\Delta x_{(i,j)(i',j')}})$ allerdings unbekannt sind, wird vorher eine Anzahl von Kombinationen für dieses Variablenpaar festgelegt. Anschließend wird für jede räumliche Kante das zeitliche MCF-Problem, das mit Gleichung (3.12) und (3.14) gegeben ist, nacheinander mit jeder der Kombinationen gelöst. Dies erfordert natürlich Kenntnis über die zu erwartenden Topographiefehler und Deformationsgeschwindigkeiten, damit für die Variablen $(\Delta z_{\Delta x_{(i,j)(i',j')}} , \Delta v_{\Delta x_{(i,j)(i',j')}})$ entsprechende Wertebereiche festgelegt werden können, aus denen die Kombinationen gebildet werden können.

Die endgültige Lösung ergibt sich dann für diejenige Kombination von $(\Delta z_{\Delta x_{(i,j)}(i',j')}, \Delta v_{\Delta x_{(i,j)}(i',j')})$, bei der die Gesamtkosten minimal werden:

$$C_{\Delta x_{(i,j)}(i',j')}|min = \sum_{\forall \Delta t_{\alpha,\beta} \in M'} \left| \tilde{k}_{\Delta x_{(i,j)}(i',j')}^{\Delta t_{\alpha,\beta}} \right|. \quad (3.15)$$

Dieser Lösung lässt sich ein optimaler Vektor von Mehrdeutigkeitsfaktoren $\tilde{\mathbf{k}}_{\Delta x_{(i,j)}(i',j')}^{\Delta t}$ zuordnen, mit dem sich die geschätzten, absoluten Phasengradienten bezüglich einer Menge lagegleicher Kanten berechnen lassen

$$\tilde{\phi}_{\Delta x_{(i,j)}(i',j')}^{\Delta t} = \tilde{\chi}_{\Delta x_{(i,j)}(i',j')}^{\Delta t} + 2\pi \tilde{\mathbf{k}}_{\Delta x_{(i,j)}(i',j')}^{\Delta t}. \quad (3.16)$$

3.3.2 Spatial Unwrapping

Wird das beschriebene Temporal Unwrapping für jede Menge lageidentischer Kanten aus der räumlichen Triangulation in der $A_Z \times R_G$ -Ebene durchgeführt, so erhält man für jede räumliche Kante in jedem differentiellen Interferogramm einen geschätzten, abgewickelten Phasengradienten $\tilde{\phi}_{\Delta x_{(i,j)}(i',j')}^{\Delta t_{\alpha,\beta}}$. Ausgehend von dieser Lösung wird nun die räumliche Phasenabwicklung auf jedem D-InSAR-Bild ausgeführt, wie in Abschnitt 3.2 dargestellt. Dabei ersetzen die geschätzten Phasengradienten $\tilde{\phi}_{\Delta x_{(i,j)}(i',j')}^{\Delta t_{\alpha,\beta}}$ die mehrdeutigen Gradienten in Formel (3.5) und es ergeben sich neue Restriktionen für das Spatial Unwrapping:

$$k_{\Delta x_{(i,j)}(i',j')}^{\Delta t_{\alpha,\beta}} + k_{\Delta x_{(i',j')(i'',j'')}}^{\Delta t_{\alpha,\beta}} + k_{\Delta x_{(i'',j'')(i,j)}}^{\Delta t_{\alpha,\beta}} = - \left[\frac{\tilde{\phi}_{\Delta x_{(i,j)}(i',j')}^{\Delta t_{\alpha,\beta}} + \tilde{\phi}_{\Delta x_{(i',j')(i'',j'')}}^{\Delta t_{\alpha,\beta}} + \tilde{\phi}_{\Delta x_{(i'',j'')(i,j)}}^{\Delta t_{\alpha,\beta}}}{2\pi} \right]. \quad (3.17)$$

Außerdem können die optimalen Gesamtkosten $C_{\Delta x_{(i,j)}(i',j')}|min$ genutzt werden, um Gewichte für die räumlichen Kanten zu berechnen. Hohe Kosten im Temporal Unwrapping Schritt bedeuten, dass die Wahrscheinlichkeit für Diskontinuitäten bei dieser räumlichen Kante höher sind. Daher sollte das Gewicht für den Mehrdeutigkeitsfaktor niedrig sein, da dieser dann einen geringen Einfluss auf die Gesamtkosten hat. PEPE UND LANARI (2006) schlagen dafür ein exponentielles Modell

$$w_{\Delta x_{(i,j)}(i',j')}^{\Delta t_{\alpha,\beta}} = \begin{cases} 2^S / 2^{C_{\Delta x_{(i,j)}(i',j')}|min} & \text{für } C_{\Delta x_{(i,j)}(i',j')}|min < \rho \text{ mit } C_{\Delta x_{(i,j)}(i',j')}|min < S \\ 1 & \text{für } C_{\Delta x_{(i,j)}(i',j')}|min \geq \rho \end{cases} \quad (3.18)$$

vor, wobei die maximalen Kosten mit 2^S festgelegt sind und ρ einen Schwellwert darstellt, der nicht größer als $0,05 \cdot M'$ gewählt werden sollte.

Diese Gewichte $w_{\Delta x_{(i,j)}(i',j')}^{\Delta t_{\alpha,\beta}}$ gehen wie folgt in die Minimierungsfunktion des Spatial Unwrappings ein:

$$\sum_{\forall \Delta x_{(i,j)}(i',j') \in M} w_{\Delta x_{(i,j)}(i',j')}^{\Delta t_{\alpha,\beta}} \left| k_{\Delta x_{(i,j)}(i',j')}^{\Delta t_{\alpha,\beta}} \right| \dots \min. \quad (3.19)$$

3.4 Der Edgelist Algorithmus

Mit dem *Edgelist*-Algorithmus von (SHANKER UND ZEBKER, 2010) wird hier ein weiteres 3D-Phase Unwrapping-Verfahren vorgestellt, das auf der Lösung eines ganzzahligen linearen Programms beruht. Allerdings soll hier nur das Grundprinzip des Algorithmus vorgestellt werden. Einen tieferen Einblick bietet SHANKER (2010). Eine verständliche Aufarbeitung des Verfahrens inklusive der Grundlagen zur linearen Programmierung ist in WALZOG (2014) zu finden.

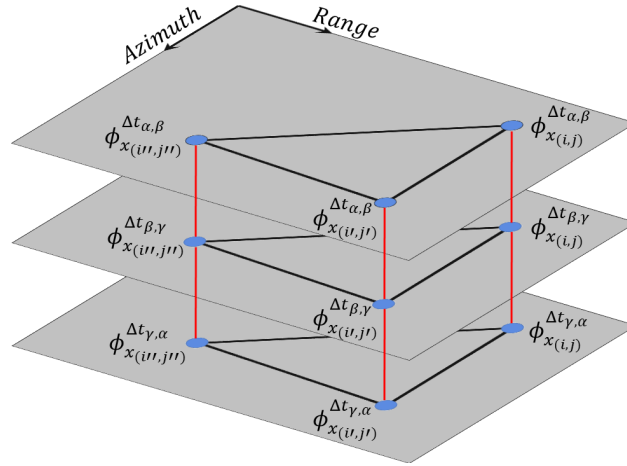


Abbildung 3.3: Darstellung der Graphenstruktur für den 3D-Fall des Edgelist-Algorithmus. Es wird zwischen räumlichen Kanten (schwarz) und zeitlichen Kanten (rot) unterschieden.

Der Edgelist-Algorithmus zielt auf die Phasenabwicklung von dünnbesetzten Daten ab und ist sowohl auf einzelne Interferogramme als auch auf Bildstapel anwendbar. Außerdem kann er im Gegensatz zum EMCF-Algorithmus für das Phase Unwrapping von Bildstapel verwendet werden, die für Persistent Scatterer-Methoden oder SBAS-Methoden erzeugt wurden (SHANKER UND ZEBKER, 2010).

Analog zur EMCF-Technik wird von einem Set von kohärenten Punkten (oder Persistent Scatterern) ausgegangen, deren Lage in jedem Interferogramm gleich ist. Diese Punkte sollen in jedem Interferogramm durch eine nicht-überlappende Vermaschung (z.B. eine Delaunay-Triangulation) verbunden sein. Das heißt, es dürfen keine Blattknoten existieren und jede Kante soll Teil einer Schleife sein. Zeitliche Kanten werden zwischen lageidentischen Knoten benachbarter Bilder im Stapel erzeugt (vgl. Abbildung 3.3).

Zunächst sei hier nochmal der Zusammenhang zwischen dem gesuchten, absoluten und dem gegebenen, mehrdeutigen Phasenwert mit

$$\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}} = \psi_{x(i,j)}^{\Delta t_{\alpha,\beta}} + 2\pi k_{x(i,j)}^{\Delta t_{\alpha,\beta}}$$

gezeigt. Darauf basierend formulieren SHANKER UND ZEBKER (2010) eine Restriktion für jede Kante zwischen zwei Pixeln $x(i,j)$ und $x(i',j)$ im selben Interferogramm über

$$k_{x(i',j')}^{\Delta t_{\alpha,\beta}} - k_{x(i,j)}^{\Delta t_{\alpha,\beta}} + k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}} = \left[\frac{\psi_{x(i,j)}^{\Delta t_{\alpha,\beta}} - \psi_{x(i',j')}^{\Delta t_{\alpha,\beta}}}{2\pi} \right] \quad (3.20)$$

bzw. für eine zeitliche Kanten zwischen zwei Interferogrammen über

$$k_{x(i,j)}^{\Delta t_{\gamma,\delta}} - k_{x(i,j)}^{\Delta t_{\alpha,\beta}} + k_{\Delta x(i,j)}^{\Delta t_{(\alpha,\beta)(\gamma,\delta)}} = \left[\frac{\psi_{x(i,j)}^{\Delta t_{\alpha,\beta}} - \psi_{x(i,j)}^{\Delta t_{\gamma,\delta}}}{2\pi} \right]. \quad (3.21)$$

Da es sich bei den Phasenwerten eines Interferogramms bereits um zeitliche Differenzen handelt, entspricht Gleichung (3.21) einer zeitlichen Doppeldifferenz. Daher treten bei dem Mehrdeutigkeitsfaktor $k_{\Delta x(i,j)}^{\Delta t_{(\alpha,\beta)(\gamma,\delta)}}$ vier Zeitpunkte auf.

Beschränkt man sich nun vorerst auf den 2D-Fall also Gleichung (3.20), so beschreiben $k_{x(i',j')}^{\Delta t_{\alpha,\beta}}$ und $k_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ Knotenpotenziale und $k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$ den Fluss auf der Kante. $[\cdot]$ steht wieder für den Rundungsoperator zum

nächsten Integer.

Wie bereits in Abschnitt 3.1 bei Gleichung (3.2) festgestellt wurde, kann der Mehrdeutigkeitsfaktor $k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$ für einen Phasengradienten um ± 1 von der Differenz der Mehrdeutigkeitsfaktoren der Phasen abweichen, die den Gradienten bilden. Dies wird durch die Bedingungen des Edgelist-Algorithmus ausgedrückt, denn die rechte Seite kann ebenfalls nur die Werte $\{-1, 0, 1\}$ annehmen. Die gleiche Argumentation stimmt ebenfalls für die zeitlichen Kanten.

Die Zielfunktion ist analog zu Abschnitt 3.2 die Minimierung der gewichteten Summe der Flussvariablen, da es sich hier auch um ein MCF-Problem handelt. Daher müssen die Mehrdeutigkeitsfaktoren für die Phasengradienten ebenfalls in ihre positiven und negativen Anteile aufgespalten und in den Restriktionen ersetzt werden (vgl. Gleichung (3.6) und (3.7)).

Im Gegensatz zur „klassischen“ Formulierung des Phase Unwrappings als MCF-Problem werden beim Edgelist Algorithmus die Mehrdeutigkeitsfaktoren $k_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ für die interferometrischen Phasen $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ mitbestimmt. Diese können direkt an die mehrdeutigen Phasenwerte $\psi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ angebracht werden, weshalb kein zusätzlicher Integrationsschritt nötig ist.

Von Nachteil ist jedoch, dass durch das Aufstellen einer Restriktion pro Kante und der Einführung der Mehrdeutigkeitsfaktoren $k_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ als zusätzliche Variablen das zu lösende System stark vergrößert wird.

3.5 Die Small BAseline Subset-Technik

In den vorangegangenen Abschnitten wurde das Phase Unwrapping inklusive einiger speziellen Methoden geschildert, da dies den ersten und kritischsten Prozessierungsschritt darstellt, der auf einen D-InSAR-Bildstapel angewendet wird, wie er für diese Arbeit vorliegt. Nun soll mit der *Small BAseline Subset* (SBAS)-Technik der abschließende Schritt erläutert werden.

Mit dieser Methode ist es möglich, die Deformationsraten der Erdoberfläche aus einem Stapel von abgewickelten, differentiellen Interferogrammen zu bestimmen (BERARDINO ET AL., 2002).

Wie in Abschnitt 2.5 bereits erörtert wurde, gibt es mit den Permanent Scatterer-Anwendungen auch andere Methoden zur Deformationsanalyse. Allerdings wurde auch gezeigt, dass der vorliegende Datenstapel mit dem Ziel einer späteren Auswertung mit der SBAS-Technik erzeugt wurde (vgl. Abschnitt 2.5.2). Aus diesem Grund beschränken wir uns hier auf die von BERARDINO ET AL. (2002) beschriebene Methode, die in ihren Grundzügen vorgestellt werden soll.

Ausgegangen wird von einem Bildstapel mit M' abgewickelten D-InSAR-Szenen, die auf eine Referenzszenen koregistriert sind. Diese wurden aus N' SAR-Bildern erzeugt, die zu den chronologisch geordneten Zeitpunkten $(t_1, \dots, t_{N'})$ aufgenommen wurden (vgl. Abschnitt 2.5.1). Der Datenstapel kann dabei aus mehreren Subsets bestehen, wie in Abbildung 2.3 gezeigt. Für die Interferogramme lassen sich nun zwei Indexvektoren

$$\mathbf{i}_{master} = [i_{1|master}, \dots, i_{M'|master}]^T \quad \mathbf{i}_{slave} = [i_{1|slave}, \dots, i_{M'|slave}]^T \quad (3.22)$$

aufstellen, welche die Indizes der Aufnahmezeitpunkte der Master- bzw. Slaveszenen im Vektor $\mathbf{t} = [t_1, \dots, t_{N'}]^T$ enthalten. Es wird angenommen, dass die Master- und Slavebilder chronologisch geordnet sind, sodass für das m -te Interferogramm beispielsweise $i_{m|master} > i_{m|slave}$, $\forall m = 1, \dots, M'$ gilt. Für eine absolute Phase in Pixel $x(i,j)$ eines Interferogramms, berechnet aus den SAR-Szenen zu den Zeiten t_α und t_β , gilt

$$\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}} = \phi_{x(i,j)}^{t_\beta} - \phi_{x(i,j)}^{t_\alpha} \quad (3.23)$$

$$\approx \frac{4\pi}{\lambda} [d(t_\beta, i, j) - d(t_\alpha, i, j)], \quad (3.24)$$

wobei $d(t_\alpha, i, j)$ und $d(t_\beta, i, j)$ die Anteile der Gesamtdeformation in Blickrichtung des Radars zu den Zeitpunkten t_α und t_β in Bezug auf den Referenzzeitpunkt t_1 sind. Damit wird implizit festgelegt, dass die Deformation zum Zeitpunkt t_1 Null ist und damit auch $\phi_{x(i,j)}^{t_1} = 0$ gilt.

Prinzipiell ließe sich mit Gleichung (3.24) für jedes Pixel $x(i,j)$ ein Gleichungssystem der Form

$$\mathbf{A}\phi_{x(i,j)}^t = \phi_{x(i,j)}^{\Delta t} \quad (3.25)$$

aufstellen, wobei \mathbf{A} die zu den Beobachtungsgleichungen (3.23) zugehörige Designmatrix ist. Die Lösung des Gleichungssystems (3.25) wären die Phasenwerte zu den Zeitpunkten t_α , die sich in Deformationen $d(t_\alpha, i, j)$ mit $\alpha = 2, \dots, N'$ umrechnen ließen. Nach BERARDINO ET AL. (2002) kann dies jedoch zu physikalisch bedeutungslosen Ergebnissen führen. Daher wird ein alternatives Gleichungssystem aufgestellt. Die Unbekannten sind die Phasengeschwindigkeiten zwischen benachbarten Aufnahmezeitpunkten

$$\mathbf{v}_{x(i,j)}^{\Delta t} = \left[v_{x(i,j)}^{\Delta t_{1,2}} = \frac{\phi_{x(i,j)}^{t_2} - \phi_{x(i,j)}^{t_1}}{t_2 - t_1}, \dots, v_{x(i,j)}^{\Delta t_{N'-1, N'}} = \frac{\phi_{x(i,j)}^{t_{N'}} - \phi_{x(i,j)}^{t_{N'-1}}}{t_{N'} - t_{N'-1}} \right]^T \quad (3.26)$$

und die Beobachtungsgleichung bezüglich der abgewickelten Phase an Pixel $x(i,j)$ im m -ten Interferogramm (bzgl. der Zeitpunkte t_α und t_β) lauten

$$\sum_{n=i_{m|slave}+1}^{i_{m|master}} (t_n - t_{n-1}) v_{x(i,j)}^{\Delta t_{n-1, n}} = \phi_{x(i,j)}^{\Delta t_{\alpha, \beta}}. \quad (3.27)$$

Dies lässt sich als ein Gleichungssystem

$$\mathbf{B}\mathbf{v}_{x(i,j)}^{\Delta t} = \phi_{x(i,j)}^{\Delta t} \quad (3.28)$$

darstellen mit der Designmatrix \mathbf{B} der Dimension $M' \times (N' - 1)$. Ihre Einträge lassen sich über

$$B(m, n) = \begin{cases} t_{n+1} - t_n & , i_{m|slave} \leq n \leq i_{m|master} - 1, \forall m = 1, \dots, M' \\ 0 & , \text{sonst} \end{cases} \quad (3.29)$$

berechnen.

Für den Datenstapel, wie in Abschnitt 2.5.1 beschrieben, gilt $N' - 1 \leq M'$, sodass das Gleichungssystem in (3.28) entweder quadratisch oder überbestimmt ist. Besteht der D-InSAR-Bildstapel lediglich aus einem Subset, so kann man das Gleichungssystem über einen Kleinste-Quadrate-Ausgleich

$$\tilde{\mathbf{v}}_{x(i,j)}^{\Delta t} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \phi_{x(i,j)}^{\Delta t} \quad (3.30)$$

lösen, da \mathbf{B} den Rang $N' - 1$ hat. Bei L Subsets (mit $L > 1$) liegt jedoch ein Rangdefekt von $L - 1$ vor, also $\text{Rang}(\mathbf{B}) = N' - L$, sodass eine Lösung über

$$\tilde{\mathbf{v}}_{x(i,j)}^{\Delta t} = \mathbf{B}^- \phi_{x(i,j)}^{\Delta t} \quad (3.31)$$

möglich ist, wobei \mathbf{B}^- für die Pseudoinverse steht. Die Pseudoinverse löst das System im Sinne der Kleinsten-Quadrate-Ausgleichung so, dass aus der Lösungsmannigfaltigkeit, die durch den Rangdefekt bedingt ist, die

Lösung mit der kleinsten euklidischen Norm gewählt wird. Aus den geschätzten Phasengeschwindigkeiten kann der Phasenwert für ein Pixel $x_{(i,j)}$ zum Zeitpunkt t_α mit

$$\phi_{x_{(i,j)}}^{t_\alpha} = \sum_{n=2}^{\alpha} \tilde{v}_{x_{(i,j)}}^{\Delta t_{n-1,n}} (t_n - t_{n-1}) \quad \forall \alpha = 2, \dots, N' \quad (3.32)$$

bestimmt werden. Nach einer Skalierung mit $\frac{\lambda}{4\pi}$ ergibt dieser die gesuchte Deformation $d(t_\alpha, i, j)$.

Die hier vorgestellte Vorgehensweise beschreibt nur die Grundidee hinter der SBAS-Methode und vernachlässigt in Formel (3.24) mehrere Phasenkomponenten, die für eine korrekte Berechnung der Deformation berücksichtigt werden müssen. Dazu gehören Abweichungen durch Fehler im DHM, die bei der Elimination der topographischen Phase entstehen, und Abweichungen durch atmosphärische Inhomogenitäten zwischen den Aufnahmezeitpunkten sowie das Phasenrauschen durch den Dekorrelationseffekt. Für eine erweiterte Darstellung mit Berücksichtigung der genannten Abweichungen sei daher auf BERARDINO ET AL. (2002) und CASU (2009, S. 55–61) verwiesen.

4. Aufbau und Funktionalitäten der Datenstrukturen

Im folgenden Kapitel sollen die in dieser Arbeit entwickelten Datenstrukturen vorgestellt werden. Hierfür werden in Abschnitt 4.1 Anforderungen an die Datenstrukturen herausgearbeitet, die sich aus den Grundlagen der Kapitel 2 und 3 ergeben.

In Abschnitt 4.2 soll der Aufbau der Strukturen präsentiert werden. Mit Hilfe eines objektorientierten Ansatzes und der grafischen Darstellung von *Unified Modeling Language*-Diagrammen (UML) werden die einzelnen Komponenten definiert und deren Abhängigkeiten zueinander skizziert.

Abschließend werden in Abschnitt 4.3 einige Anfragen an die Datenstrukturen vorgeschlagen, die von dieser unterstützt werden sollten. Diese zielen darauf ab, dass die Struktur zur Untersuchung und Auswertung von D-InSAR-Bildstapeln verwendet werden kann.

4.1 Anforderungen an die Datenstrukturen

Bereits zu Beginn dieses Abschnittes ist es notwendig, darauf hinzuweisen, dass Datenstrukturen zur Beschreibung eines D-InSAR-Bildstapels niemals vollständig sein können. Es gibt eine Vielzahl von Anwendungen im Bereich der differentiellen SAR-Interferometrie. Insbesondere der Bereich des Phase Unwrappings stellt die unterschiedlichsten Anforderungen an eine Datenstruktur. Folglich wird es stets Anforderungen an den Informationsgehalt sowie an die Funktionalitäten geben, die eine Datenstruktur nicht erfüllen kann.

Aus diesem Grund wird nun das Ziel verfolgt, eine möglichst vollständige Beschreibung der Daten in Hinblick auf eine bestimmte Menge von Anwendungen zu realisieren. Die Struktur sollte dabei jedoch so flexibel sein, dass in einem gewissen Maße Erweiterungen für andere Anwendungen möglich sind.

Für die hier entwickelten Datenstrukturen sind die in Kapitel 2 und 3 vermittelten Grundlagen und Anwendungen sowie die vorliegenden Daten (vgl. Abschnitt 2.5.2) maßgebend. Dies schränkt die Menge an abzudeckenden Applikationen bereits ein. In erster Instanz bedeutet dies, dass nur solche Anwendungen unterstützt werden, die auf dünnbesetzten Interferogrammen aufbauen. Des Weiteren werden nur Datenstapel unterstützt, die für die Auswertung mit der SBAS-Methode erzeugt wurden.

Bereits in Abschnitt 2.5 wurde gezeigt, dass sich der zeitliche Zusammenhang interferometrischer Daten als Graph in der Basislinienebene darstellen lässt. Außerdem wurde erläutert, wie dieser Graph für das Phase Unwrapping genutzt werden kann (vgl. Abschnitt 3.3). Aus diesem Grund werden zwei Substrukturen zur Beschreibung der kohärenten Pixel im D-InSAR-Bildstapel aufgebaut. Diese Substrukturen lassen sich in eine *Zeitliche* sowie *Räumliche Struktur* aufgliedern. Die Zeitliche Struktur basiert dabei auf der Modellierung des Graphen in der Basislinienebene. Die Räumliche Struktur dagegen spiegelt den Graphen wieder, der die kohärenten Pixel in den Interferogrammen verknüpft.

Die gewählte Nomenklatur orientiert sich an den Bezeichnungen des EMCF-Algorithmus (vgl. Abschnitt 3.3). Eigentlich sind die gewählten Bezeichnungen nicht einwandfrei. Die Zeitliche Struktur weist durch ihre Abhängigkeit von der orthogonalen Basislinie auch eine räumliche Komponente auf.

In den vorangegangenen Kapiteln wurde gezeigt, dass Graphen bzw. Netzwerke eine entscheidende Rolle bei der Darstellung und Prozessierung der Daten spielen, weshalb die Strukturen als Knoten und Kanten modelliert werden. Des Weiteren werden auch orientierte Flächen in der Datenstruktur abgebildet, mit denen die Berechnung von Restriktionen beim Phase Unwrapping (vgl. Gleichung (3.5)) möglich sind. Während

bisher stets von überlappungsfreien Dreiecken ausgegangen wurde, sollen die Bedingungen an die modellierten Flächen teilweise etwas aufgeweicht werden. So soll es möglich sein, beliebige Vielecke darzustellen, denn die Restriktionen lassen sich dahingehend einfach erweitern.

4.2 Modellierung der Datenstrukturen

In diesem Abschnitt soll das Modell der entwickelten Datenstruktur vorgestellt werden. Es wird ein objektorientierter Ansatz gewählt, da dies den Übergang von den anschaulichen Graphen zur abstrakteren Datenstruktur erleichtert. Außerdem erlaubt die objektorientierte Programmierung eine flexible Erweiterung des Informationsgehalts und der Funktionalitäten der Datenstruktur, indem den Klassen neue Attribute und Methoden hinzugefügt werden.

Daher sollen zuerst die modellierten Klassen und deren Beziehungen beschrieben werden. Im Anschluss wird auf den Aufbau der einzelnen Klassen eingegangen. Dazu wird für jede Klasse ein ausführliches UML-Diagramm mit den vorgesehenen Attributen und Methoden präsentiert und diskutiert.

4.2.1 Modell des Bildstapels

In Abbildung 4.1 sind die modellierten Klassen und deren Abhängigkeiten mit Hilfe eines vereinfachten UML-Diagramms dargestellt. Aus Gründen der Übersichtlichkeit wird vorerst auf die Nennung der Attribute und Methoden verzichtet. Diese werden jedoch in den Abschnitten 4.2.2, 4.2.3 und 4.3 vorgestellt.

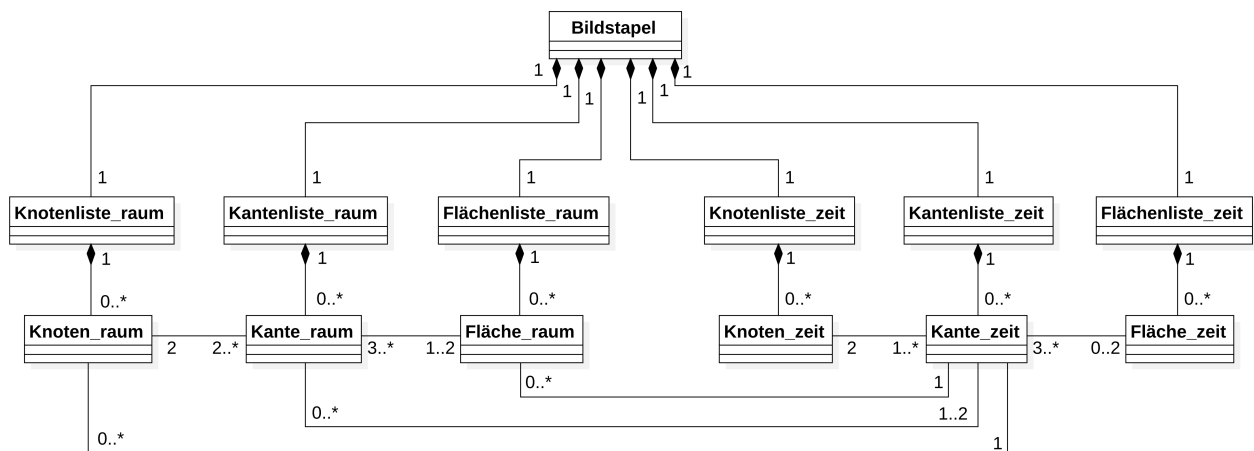


Abbildung 4.1: UML-Diagramm der modellierten Datenstruktur als Überblick über die modellierten Klassen und deren Beziehungen inklusive der Multiplizitäten. Für die Übersichtlichkeit wurde auf die Darstellung der Attribute und Methoden verzichtet.

Dem Ziel dieser Arbeit folgend wird eine übergeordnete Klasse *Bildstapel* erzeugt, die den D-InSAR-Bildstapel repräsentiert. Sie dient hauptsächlich der Verwaltung von Anfragen (vgl. Abschnitt 4.3). Damit stellt sie die Anwenderschnittstelle dar, über die der Nutzer die Datenstruktur aufbauen, ändern sowie auf deren Informationen zugreifen kann. Dies hat den Vorteil, dass klassenübergreifende Anfragen gestellt werden können und der Nutzer nicht über deren Verknüpfung nachdenken muss.

Ihr untergeordnet sind zwei Blöcke von Klassen. Auf der linken Seite in Abbildung 4.1 sind die Klassen zu finden, die die Räumliche Struktur bilden, auf der rechten Seite die Klassen der Zeitlichen Struktur.

Die Zeitliche Struktur ist ein Modell für die Basisliniennetzwerke, die in Abschnitt 2.5.1 vorgestellt wurden. Ein Knoten in diesem Graph entspricht einem SAR-Bild und eine Kante einem Interferogramm. Da in der

gegebenen Datengrundlage für jedes Interferogramm ein zugehöriges Intensitäts- und Kohärenzbild vorhanden ist, können diese als Zusatzinformationen der Kante zugeordnet werden. Die Flächen der Zeitlichen Struktur entsprechen keinem Datenprodukt. Sie werden im Hinblick auf die Nutzung der Datenstruktur für den EMCF-Algorithmus modelliert.

Außerdem wird für jedes dieser drei Graphenelemente eine Listenklasse modelliert, die der Verwaltung der Instanzen dient. Mit ihnen sollen beispielsweise Nachbarschaftsanfragen realisiert werden oder Manipulationen des Graphen, die mehr als ein Objekt betreffen. Darunter fallen unter anderem das Löschen von Kanten oder das Vereinigen von Flächen.

Die Knoten der Räumlichen Struktur repräsentieren in erster Linie die kohärenten Pixel im Bildstapel. Ihnen können jedoch weitere Informationen, wie ein Kohärenz- und Intensitätswert, zugewiesen werden. Somit stellt die Räumliche Struktur eine dreidimensionale Knotenwolke dar. Die Knoten in dieser Wolke sind in Ebenen angeordnet, die den D-InSAR-Bildern im Stapel entsprechen. Mit Hilfe einer Assoziation zwischen der Klasse *Knoten_raum* und der Klasse *Kante_zeit* wird dieser Zusammenhang modelliert.

Zur Verknüpfung der Knoten untereinander wird eine Kantenklasse erzeugt, die für die Phasengradienten steht. Hier wird zwischen Kanten unterschieden, die innerhalb einer Ebenen liegen, und solchen, die zwischen Knoten benachbarter Ebenen aufgespannt werden. Die erste Gruppe entspricht demnach den Phasengradienten in einem Interferogramm und die zweite den Kanten, die beim Edgelist-Algorithmus als zeitliche Kanten bezeichnet werden (vgl. Abbildung 3.3 und Gleichung (3.21)). Diese sind jedoch unbedingt von den Kanten der Zeitlichen Struktur zu unterscheiden. Der Bezug zu den D-InSAR-Bildern wird wiederum durch eine Assoziation zur Klasse *Kante_zeit* umgesetzt.

Als letztes Basiselement der Räumlichen Struktur werden Flächen modelliert, die als Phasenzyklen aufgegriffen werden können. Wie der Assoziation zur Klasse *Kante_zeit* zu entnehmen ist, werden nur Flächen in der Struktur abgebildet, die innerhalb von Interferogrammen liegen.

Auch die Elemente dieses Netzwerkes werden durch Listenklassen verwaltet, die weitestgehend die gleichen Aufgaben haben, wie die Listen der Zeitlichen Struktur. Zusätzlich sollen sie den Zugriff auf Objekte erleichtern, die in einem gemeinsamen Bildebene liegen, oder auf Objekte, die die gleiche Lagen entlang des Bildstapels haben (z.B. Knoten mit identischen Bildkoordinaten).

Wie bereits beschrieben, sind die beiden Strukturen über die Klasse *Kante_zeit* miteinander verknüpft. Die Assoziation zwischen dieser Klasse und den Kanten der Räumlichen Struktur erlaubt es, Zuordnungen durchzuführen, wie sie für das Temporal Unwrapping beim EMCF-Algorithmus nötig sind (vgl. Abbildung 3.2 und Abschnitt 3.3.1).

4.2.2 Modell der Zeitlichen Struktur

Der Graph der zeitlichen Struktur, also das Basisliniennetzwerk (vgl. Abschnitt 2.5.1), besteht aus den Klassen *Knoten_zeit*, *Kante_zeit* und *Fläche_zeit*, die wiederum in den korrespondierenden Listen *Knotenliste_zeit*, *Kantenliste_zeit* sowie *Flächenliste_zeit* verwaltet werden.

Ausgegangen wird von einem Bildstapel, wie er für den EMCF-Algorithmus benötigt wird (siehe Abbildung 2.4), also von einer Dreiecksvermaschung in der Basislinienebene. Es soll jedoch möglich sein, D-InSAR-Szenen aus dem Bildstapel zu entfernen bzw. in diesen einzufügen. Somit ist die Verallgemeinerung zu einem beliebigen SBAS-Bildstapel realisierbar. Diese Änderungen entsprechen dem Löschen bzw. Einfügen von Kanten in der Zeitlichen Struktur und haben somit Auswirkungen auf den Graphen.

Das Entfernen einer Kante soll kurz an den zwei Minimalbeispielen in Abbildung 4.2 verdeutlicht werden. Gelöscht wird jeweils die rote, gestrichelte Kante. In Abbildung 4.2a grenzt an eine solche Kante auf *bei-*

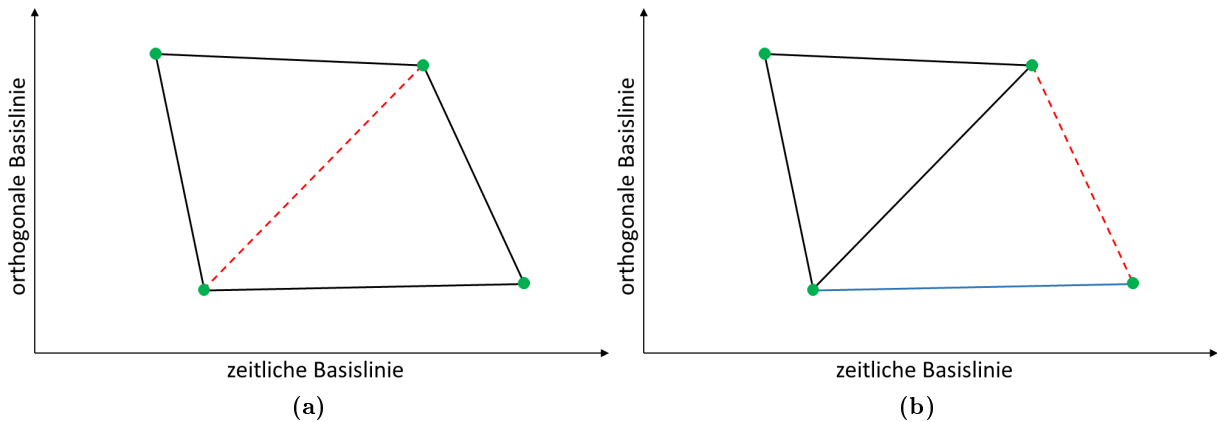


Abbildung 4.2: Konsequenzen des Löschs einer Kante für den Graphen.

den Seiten eine Masche an.¹ Durch Elimination dieser Kante entsteht eine neue Masche, die kein Dreieck ist. Im UML-Diagramm ist dies durch die Multiplizität 3..* an der Assoziation zwischen *Kanten_zeit* und *Flächen_zeit* angedeutet. Diese Assoziation definiert, dass eine Fläche von mindestens drei Kanten begrenzt wird.

In Abbildung 4.2b wird eine Kante gelöscht, die nur auf *einer* Seite von einer Maschen begrenzt wird.² Dies kann zur Folge haben, dass eine oder mehrere Kanten (hier in blau dargestellt) von *keiner* Masche begrenzt werden (bzw. auf beiden Seiten von der offenen Masche). Anhand der Multiplizität 0..2 der bereits oben genannten Assoziation kann dies nachvollzogen werden. Dadurch wird die Forderung deutlich, dass an eine Kante mindestens null, höchstens jedoch zwei Flächen angrenzen. Außerdem kann ein Knoten demnach nur Teil von einer Kante sein, was bei durchgängig geschlossenen Zyklen nicht der Fall sein kann.

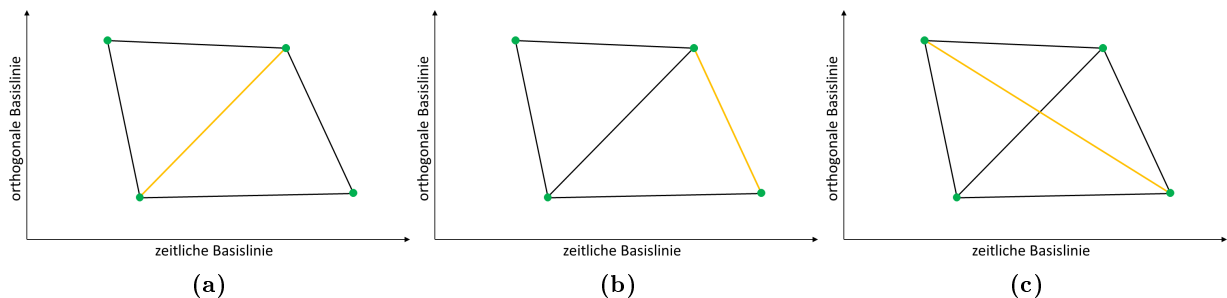


Abbildung 4.3: Konsequenzen des Einfügens einer Kante für den Graphen. Die gelbe Kante wird jeweils in einen bestehenden Graphen eingefügt.

Wird eine Kante hinzugefügt, können drei Fälle auftreten. Diese sind in Abbildung 4.3 dargestellt, wobei stets die gelbe Kante eingefügt wird. Entweder wird eine Fläche in zwei Teilflächen unterteilt (vgl. Abbildung 4.3a) oder ein Zyklus wird wieder geschlossen (vgl. wie in Abbildung 4.3b). Der letzte Fall führt zu sich überschneidenden Kanten (vgl. Abbildung 4.3c).

Der erste Fall in Abbildung 4.3a wird in der hier erzeugten Datenstruktur berücksichtigt, indem nach dem Einfügen die alte Fläche in zwei neue Maschen unterteilt wird. In den anderen beiden Fällen ist es jedoch nicht gelungen, die angrenzenden Flächen an der eingefügten Kante zu erzeugen. Das Problem liegt hier in der Identifikation der richtigen Zyklen und konnte nicht zufriedenstellend gelöst werden. Wenn also der erste Fall nicht auftritt, wird lediglich eine Kante erzeugt, deren Anrainer auf die Masche *Außen* verweisen.

¹ Die Flächen, die an eine Kante grenzen, werden auch als Anrainer bezeichnet.

² Oftmals wird angenommen, dass eine Kante stets von zwei Maschen begrenzt wird. Für Kanten der konvexen Hülle ist einer der Anrainer die fiktive, offene Masche, die häufig als *Außen* bezeichnet wird.

Bis hierhin wurde gezeigt, welche Erfordernisse an die Zeitliche Struktur gestellt werden. Nun wird explizit gezeigt, wie die Klassen aufgebaut werden, damit dies umgesetzt werden kann. Dazu wird für jede Klasse der Zeitlichen Struktur ein ausführliches UML-Diagramm präsentiert. Anhand dieser sollen die gewählten Attribute erläutert sowie Methoden vorgeschlagen werden, die die Klassen besitzen sollten.

Knotenliste_zeit

Als erste der Verwaltungsstrukturen wird nun die Klasse *Knotenliste_zeit* vorgestellt. Die Abbildung 4.4 zeigt das zugehörige UML-Diagramm.

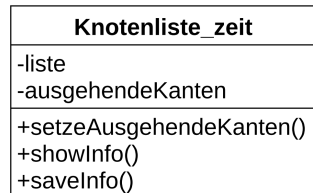


Abbildung 4.4: UML-Diagramm der Klasse *Knotenliste_zeit*.

In dieser Klasse werden die Knotenobjekte der Zeitlichen Struktur in einer Liste gespeichert. Außerdem wird eine Referenzstruktur erzeugt, in der für jeden Knoten eine Liste mit den IDs der Kanten eingetragen wird, die von diesem Knoten ausgehen oder an ihm enden. So ist es möglich, effektiv Nachbarschaftsanfragen zu stellen.

liste ... Liste mit den Knotenobjekten der Zeitlichen Struktur.
ausgehendeKanten ... Referenzen auf die ausgehenden (bzw. ankommenden) Kanten für jeden Knoten.

Die Methode *setzeAusgehendeKanten()* erzeugt die Liste mit den eben erwähnten Referenzen. Diese kann nicht direkt vom Konstruktor erzeugt werden, da hierfür die Kanten in der Zeitlichen Struktur benötigt werden und diese erst nach Erzeugung der Punktliste instanziiert werden können. Ausgehend von der Liste der Kanten lässt sich die Referenzstruktur einfach aufbauen, indem für jede Kante deren ID in derjenigen Liste eingetragen wird, die dem Anfangs- und Endknoten zugeordnet ist.

Neben dem Speichern und Verknüpfen der Informationen soll weiterhin ein geordneter Abruf der Daten ermöglicht werden. Dies ist mit den Methoden *showInfo()* und *saveInfo()* möglich, die jeder Listenklasse zu eigen sind. Mit *showInfo()* können alle Informationen bezüglich eines spezifischen Objektes, das über seine ID identifiziert wird, in einer Konsole ausgegeben werden. Die Methode *saveInfo()* speichert für alle Objekte in der Liste die gleichen Informationen in einer Textdatei. Für diese Klasse umfasst die Ausgabe alle Attribute der Klasse *Knoten_zeit* sowie die Liste mit den IDs der ausgehenden Kanten.

Knoten_zeit

Die Knoten in der Basislinienebene werden durch die Klasse *Knoten_zeit* repräsentiert, deren Modell dem UML-Diagramm in Abbildung 4.5 entspricht.

Jeder Knoten soll eine eindeutige ID haben, über die dieser identifiziert und auf ihn zugegriffen werden kann. Den Ausführungen aus Abschnitt 2.5 folgend wird für jeden Knoten die zeitliche Basislinie Δt_α bezüglich der gewählten Referenzszene als Attribut gespeichert. Gleiches gilt für die orthogonal Basislinie $b_{\perp|\alpha}$ und die Doppler-Frequenz-Verschiebung der SAR-Szene $f_{c|\alpha}$. Für einen absoluten Zeitbezug wird außerdem das Aufnahmedatum der SAR-Szene als weiteres Attribut gewählt. Beispielsweise benötigt die SBAS-Methode für die chronologische Ordnung der Indexvektoren aus Gleichung (3.22) einen solchen absoluten Zeitbezug.

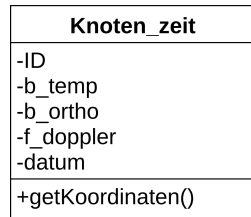


Abbildung 4.5: UML-Diagramm der Klasse *Knoten_zeit*.

Die Knoteneigenschaften werden den Attributen aus dem UML-Diagramm wie folgt zugewiesen:

- ID** ... Die eindeutige ID für die Identifikation und den Zugriff auf den Knoten.
b_temp ... Die zeitliche Basislinie Δt_α bezüglich der Referenzszene.
b_ortho ... Die orthogonale Basislinie $b_{\perp|\alpha}$ bezüglich der Referenzszene.
f_doppler ... Die Doppler-Frequenz-Verschiebung der SAR-Szene $f_{c|\alpha}$.
datum ... Das Aufnahmedatum der SAR-Szene.

Generell gilt es, die Standard Getter- und Setter-Methoden zu implementieren, die das Auslesen bzw. das Setzen der einzelnen Attribute erlauben. Diese werden allerdings aus Gründen der Übersichtlichkeit in den UML-Diagrammen nicht explizit dargestellt.

Zusätzlich wird vorgeschlagen, eine Methode *getKoordinaten()* zu implementieren, welche die Koordinaten des Knotens in der Basislinienebene zurückgibt.

Kantenliste_zeit

Die Verwaltungsklasse für die Kanten der Zeitlichen Struktur heißt *Kantenliste_zeit* und wird nach dem UML-Diagramm in Abbildung 4.6 modelliert.

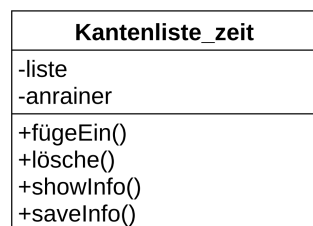


Abbildung 4.6: UML-Diagramm der Klasse *Kantenliste_zeit*.

Das entscheidende Attribut ist auch hier eine Liste mit den zu dieser Klasse korrespondierenden Objekten des Graphen in der Basislinienebene, also den Instanzen der Klasse *Kanten_zeit*. Da die Kanten in dieser Liste den differentiellen Interferogrammen entsprechen, gibt die Reihenfolge, in der die Objekte in ihr abgelegt sind, auch die Anordnung des Bildstapels vor. Daher sollte bei der Erzeugung der Liste eine Sortierung ermöglicht werden. Hier bietet sich beispielsweise eine chronologische Sortierung nach den Aufnahmedaten der Slaveszenen und anschließend nach denen der Masterszene an oder auch umgekehrt.

Diese Klasse wird ebenfalls um eine Referenzstruktur ergänzt. In dieser sollen für jede Kante die Referenzen auf die Flächen gespeichert werden, die links und rechts an ihr angrenzen. Sofern eine Kante auf einer oder beiden Seiten an die fiktive Fläche *Außen* angrenzt, wird dies durch einen Platzhalter gekennzeichnet.

- liste** ... Liste mit den Kantenobjekten der Zeitlichen Struktur.
anrainer ... Liste mit Referenzen auf die angrenzenden Flächen für jede Kante.

Wie schon in der Einleitung dieses Abschnitts erläutert, stellt das Einfügen und Löschen von Kanten einen wichtigen Bestandteil zur Änderung der zeitlichen Struktur dar. Dieser Tatsache wird durch die Methoden *fügeEin()* und *lösche()* Rechnung getragen.

Durch die Methoden *showInfo()* und *saveInfo()* wird erneut ermöglicht, die Informationen zu einem Objekt auszugeben, bzw. zu allen Objekten in einer Datei geordnet zu speichern. Die Methoden geben dabei für eine Instanz der Klasse *Kante_zeit* dessen Attribute sowie die zugehörigen IDs der Anrainer aus.

Kante_zeit

Die Kanten in der Basislinienebene werden durch das UML-Diagramm der Klasse *Kante_zeit* in Abbildung 4.7 repräsentiert.

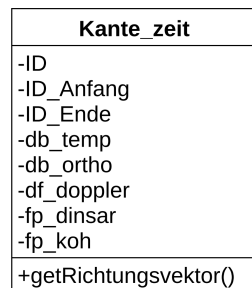


Abbildung 4.7: UML-Diagramm der Klasse *Kante_zeit*.

Auch für die Kanten gilt, dass die Identifikation und der Zugriff über einen eindeutigen Bezeichner ermöglicht werden sollen. Da eine Kante von zwei Knoten aufgespannt wird, muss der Bezug zu diesen hergestellt werden. Dies wird durch zwei Attribute realisiert, welche die ID des Anfangs- bzw. des Endknotens speichern. Wie bereits ausgeführt, lässt sich eine Kante in der Basislinienebene mit einem differentiellen Interferogramm assoziieren. Aus diesem Grund kann durch die Interferogrammerzeugung vorgegeben werden, welcher Knoten als Anfang oder Ende der Kante festgelegt werden soll. Hier wird als Startknoten derjenige Knoten aufgefasst, der der Masterszene des D-InSAR Bildes entspricht. Nach dieser Festlegung korrespondiert der Endknoten mit der Slaveszene. Somit besitzt die Kante eine vordefinierte Richtung.

Des Weiteren werden auch die relativen Basislinien $\Delta t_{\alpha,\beta}$ und $\Delta b_{\perp|\alpha,\beta}$ sowie die Doppler-Frequenz-Differenz $\Delta f_{c|\alpha,\beta}$ als Attribute dieser Klasse festgelegt. Wie in Abschnitt 2.4 erläutert wurde, sind diese Größen qualitätsbestimmend für ein D-InSAR-Bild. Damit stellen sie wichtige Eigenschaften dar, die in der Datenstruktur als Information enthalten sein sollten. Außerdem werden die relativen Basislinien für das Temporal Unwrapping im EMCF-Algorithmus benötigt.

Abschließend sind zwei Attribute für Links zu den Bildern vorgesehen, die mit der Kante assoziiert werden. Somit ist stets ein direkter Bezug zu den Daten vorhanden. Dabei sind in der Datengrundlage das Intensitäts- und Phasenbild einer D-InSAR-Szene als zweilagiges Bild gespeichert. Das zugehörige Kohärenzbild ist in einer separaten Datei enthalten.

Eine Erläuterung der Attribute aus dem UML-Diagramm in Abbildung 4.7 liefert die folgende Auflistung:

- ID** ... Die eindeutige ID für die Identifikation und den Zugriff auf die Kante.
- ID_Anfang** ... ID des Startknotens der Kante, der der Masterszene entspricht.
- ID_Ende** ... ID des Endknotens der Kante, der der Slaveszene entspricht.
- db_temp** ... Die relative zeitliche Basislinie $\Delta t_{\alpha,\beta}$ zwischen Master- und Slaveszene.
- db_ortho** ... Die relative orthogonale Basislinie $\Delta b_{\perp|\alpha,\beta}$ zwischen Master- und Slaveszene.
- df_doppler** ... Die Differenz der Doppler-Zentroiden $\Delta f_{c|\alpha,\beta}$ zwischen Master- und Slaveszene.

fp_dinsar ... Link zur Bilddatei, die das Intensitäts- und Phasenbild enthält.
fp_koh ... Link zum Kohärenzbild.

Auch für diese Klasse sollten die Standard Getter- und Setter-Methoden implementiert werden. Des Weiteren ist eine Methode vorgesehen, die den Richtungsvektor der Kanten zurückgibt. Diese geometrische Abfrage wird beispielsweise zur Bestimmung der Orientierung der Flächen benötigt.

Flächenliste_zeit

Abgeschlossen wird die Modellierung der Listen der Zeitlichen Struktur durch die Klasse *Flächenliste_zeit*, deren UML-Diagramm in Abbildung 4.8 dargestellt wird.

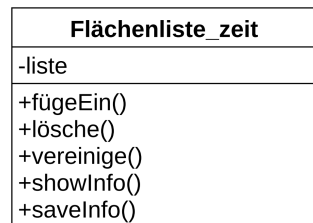


Abbildung 4.8: UML-Diagramm der Klasse *Flächenliste_zeit*.

Das einzige Attribut dieser Klasse ist eine Liste mit den Objekten der Klasse *Flächen_zeit*.

liste ... Liste mit den Flächenobjekten der Zeitlichen Struktur.

Ergänzt wird die Klasse um einige Methoden zur Verwaltung dieser Liste. Mit *fügeEin()* kann eine einzelne Fläche in die Liste eingefügt werden. Dies wird nötig, wenn durch das Einfügen einer Kante eine Masche in zwei Teilflächen zerlegt wird (vgl. Abbildung 4.3a).

Die Vereinigung zweier Maschen kann mit *vereinige()* durchgeführt werden und wird erforderlich, wenn durch das Löschen einer Kante aus zwei Teilflächen eine neue, größere Fläche entsteht (vgl. Abbildung 4.2a).

Das Löschen einer Fläche über *lösche()* wird in beiden Fällen benötigt, um die Flächen aus der Liste zu entfernen, die vor diesen Änderungen existierten.

Letztlich sollte diese Klasse ebenfalls Methoden bereitstellen, mit denen die Informationen der Teilobjekte der Liste ausgegeben werden können. Dazu dienen erneut die Methoden *showInfo()* und *saveInfo()*, die entweder die Attribute einer einzelnen Fläche in der Konsole ausgeben oder aller Flächen in eine Textdatei schreiben.

Fläche_zeit

Als letztes grundlegendes Objekt des Graphen in der Basislinienebene wird die Klasse *Fläche_zeit* modelliert, deren UML-Diagramm in Abbildung 4.9 dargestellt wird.

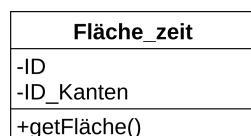


Abbildung 4.9: UML-Diagramm der Klasse *Fläche_zeit*.

Eine Fläche der Zeitlichen Struktur hat ebenfalls eine eindeutige ID für die Identifikation und den Zugriff.

Außerdem muss ein Bezug zu den Objekten hergestellt werden, die die Fläche begrenzen. Hier wird bewusst eine Liste mit Referenzen auf die begrenzenden Kanten gewählt. Oftmals werden allerdings auch die Knoten verwendet, die die Eckpunkte der Flächen bilden.

Hier werden nun die Kanten bevorzugt, da diesen ein Phasengradient zugeordnet werden kann (vgl. Abschnitt 3.3). Die Berechnung von Phasenzyklen wird somit erleichtert. Durch die Interferogrammerzeugung sind allerdings die Richtungen der Kanten vorgegeben, sodass diese nicht zwingend einen Zyklus bilden. Daher ist es zusätzlich erforderlich, eine Richtungsinformation für jede Kante zu speichern. Durch diese Zusatzinformation wird festgelegt, wie jede Kante durchlaufen werden muss, um einen Zyklus zu bilden. Dies lässt sich beispielsweise umsetzen, indem ein Vorzeichen für die IDs festgelegt wird. Ein positives Vorzeichen bedeutet, dass die Kante in der Richtung durchlaufen wird, die durch das D-InSAR Bild vorgegeben ist. Entsprechend folgt für ein negatives Vorzeichen die Traversierung der Kanten entgegen ihrer Festlegung. Außerdem werden hier die Vorzeichen so gewählt, dass alle Flächen die gleiche (positive) Orientierung haben. Dies sei an einem Minimalbeispiel in Abbildung 4.10 veranschaulicht.

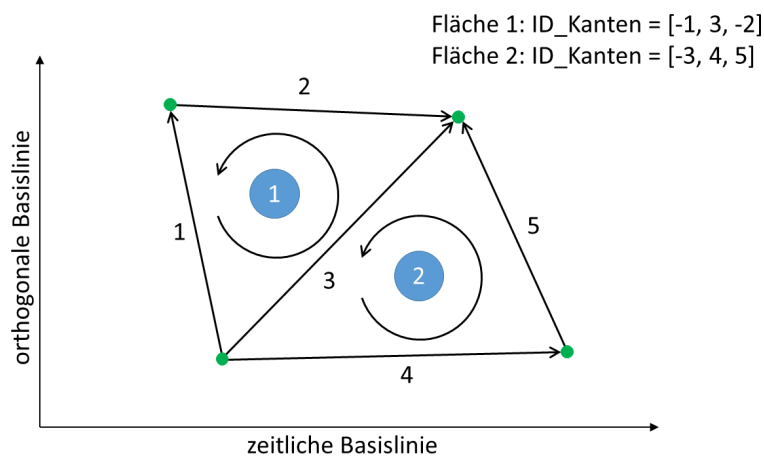


Abbildung 4.10: Speicherung der Kanten IDs für positiv orientierte Flächen.

Die Attribute im UML-Diagramm der Klasse *Fläche_zeit* sind wie folgt zu interpretieren:

ID ... Die eindeutige ID für die Identifikation und den Zugriff auf die Fläche.

ID_Kanten ... Liste der IDs der Kanten, die die Fläche begrenzen; mit Vorzeichen für die Orientierung.

Als Methoden sind wiederum die Standard Getter und Setter vorgesehen. Zudem wird eine Methode *getFläche()* implementiert, die den (orientierten) Flächeninhalt eines Dreiecks zurückgibt. Diese wird zur Identifizierung der Zyklen beim Aufbau der Zeitlichen Struktur benötigt.

Um einen Zyklus zu finden, wird eine Nachbarschaftsanfrage am Anfangs- und Endknoten einer Kante durchgeführt. Haben zwei Kanten an diesen Knoten einen gemeinsamen Punkt, so bilden diese zusammen mit der Kante, die untersucht wird, einen Zyklus. Bei einer ungünstigen Konfiguration können jedoch zwei Anrainer auf einer Seite einer Kante gefunden werden. Mit Hilfe des Flächeninhalts kann bestimmt werden, welcher der Anrainer über einen kleineren Zyklus verfügt. Diese Situation soll durch das Minimalbeispiel in Abbildung 4.11 verdeutlicht werden.

Hier wird der linke Anrainer der roten Kante gesucht. Am Anfangs- (A) und Endknoten (E) werden in der Liste der ausgehenden (bzw. ankommenden) Kanten (vgl. Klasse *Knotenliste_zeit*) diejenigen gesucht, die einen gemeinsamen Knoten haben. Bei der hier abgebildeten Konstellation werden folglich zwei Kantenpaare gefunden, einerseits die gelben mit dem gemeinsamen Knoten (G_1) und andererseits die blauen mit dem gemeinsamen Knoten (G_2). Der korrekte Anrainer mit dem kleineren Zyklus ist das Dreieck $\triangle AEG_1$.

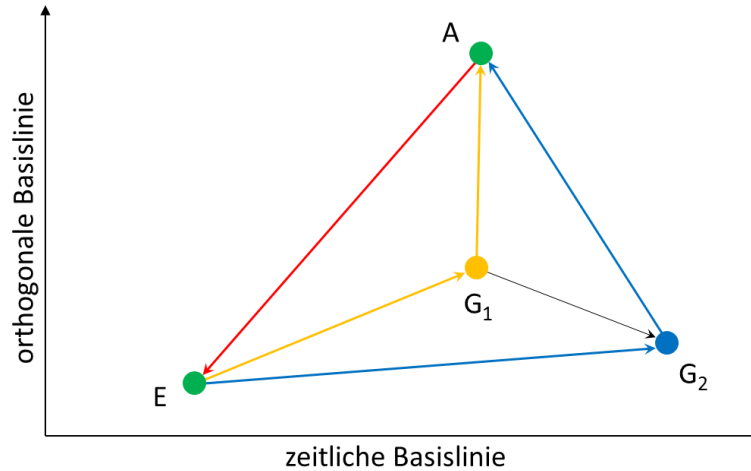


Abbildung 4.11: Sonderfall bei der Zyklensuche in der Zeitlichen Struktur.

Zur Identifikation könnten die Summen der Kantenlängen der Dreiecke verglichen werden. Hier wird jedoch der Flächeninhalt als Kriterium bevorzugt, da sich über den orientierten Flächeninhalt ebenfalls überprüfen lässt, ob der gefundene Zyklus die gewünschte positive Drehrichtung hat.

4.2.3 Modell der Räumlichen Struktur

Analog zur Zeitlichen Struktur besteht die Räumliche Struktur aus sechs Klassen. Die Klassen *Knoten_raum*, *Kante_raum* und *Fläche_raum* modellieren die Elemente des Räumlichen Graphen. Diese werden wiederum in den Klassen *Knotenliste_raum*, *Kantenliste_raum* und *Flächenliste_raum* verwaltet.

Die Auswahl der Bildpunkte, für die Knoten erzeugt werden, wird nach dem Vorgehen durchgeführt, das in Abschnitt 3.3 für den EMCF-Algorithmus vorgeschlagen wurde. Es werden somit erst Kandidaten anhand der Kohärenzbilder gesucht, die dort einen Schwellwert überschreiten. Anschließend wird für jede Pixelkoordinate die zugehörige Bildsäule betrachtet und geprüft, ob die Anzahl der Kandidaten in dieser Säule einen vorgegebenen Prozentsatz erreicht. Werden beide Kriterien erfüllt, so gehen alle Bildpunkte mit den betrachteten Pixelkoordinaten als Knoten in die Datenstruktur ein.

Zur Vermaschung der Knoten in den Ebenen der Knotenwolke ist im ersten Schritt eine Delaunay-Triangulation vorgesehen. Die gerade beschriebenen Auswahlkriterien für die Knoten haben somit den Vorteil, dass nicht für jedes Bild eine Triangulation durchgeführt werden muss. Es reicht also aus, eine Dreiecksvermaschung auf den Pixelkoordinaten zu erzeugen, die alle Knoten gemein haben. Das Ergebnis lässt sich anschließend in alle Ebenen übertragen.

Auch in der Räumlichen Struktur soll eine Nachbearbeitung der Vermaschung durch das Löschen und Einfügen von Kanten möglich sein. Die Optionen sind im Gegensatz zur Zeitlichen Struktur etwas eingeschränkter. Dies soll ebenfalls anhand der Minimalbeispiele in Abbildung 4.12 erläutert werden.

Das Löschen von Kanten aus dem Innern der Vermaschung (rot gestrichelte Kante), wie in Abbildung 4.12a dargestellt, soll auch von der Räumlichen Struktur unterstützt werden. Somit können durch Nachbearbeitung auch Vielecke entstehen. Sobald eine Kante aus einer Ebenen entfernt wird, muss diese Änderung auch in allen anderen Ebenen der Räumlichen Struktur durchgeführt werden. Auf diese Weise soll eine identische Vermaschung in allen Ebenen sichergestellt werden.

Allerdings ist der zweite Fall, der in Abbildung 4.12b gezeigt wird, an dieser Stelle anders zu bewerten als in Abschnitt 4.2.2. Hier wird die rot gestrichelte Kante gelöscht, die Teil der konvexen Hülle ist. In der Räumlichen Struktur hat dies jedoch zur Folge, dass auch die blaue Kante eliminiert wird. Folglich sind nur Kanten

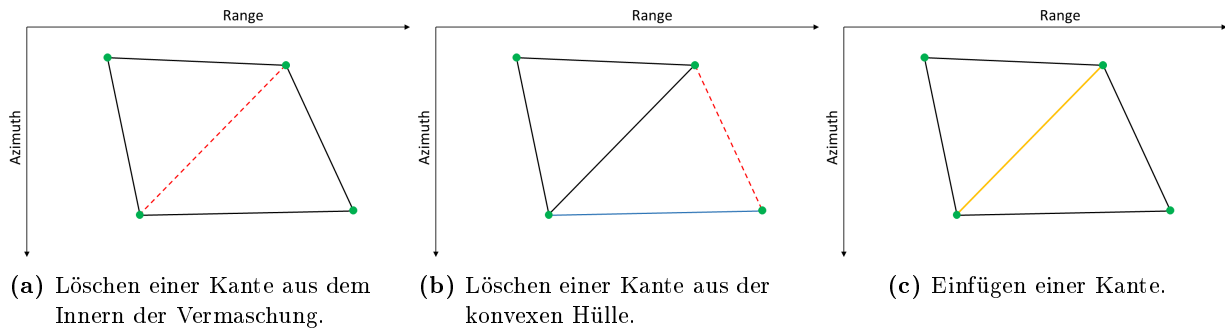


Abbildung 4.12: Löschen und Einfügen einer Kante in die Räumliche Struktur.

vorgesehen, die zumindest auf einer Seite an eine Fläche angrenzen. Dies ist ein notwendiges Kriterium für das Phase Unwrapping einzelner Interferogramme, wie es vom Spatial Unwrapping durchgeführt wird (vgl. Abschnitt 3.3.2). Hieraus ergibt sich außerdem, dass jeder Knoten Teil von mindestens zwei Kanten ist.

Diese geometrischen Verhältnisse zwischen Knoten, Kanten und Flächen werden auch durch die Multiplizitäten an den entsprechenden Assoziationen im UML-Diagramm in Abbildung 4.1 verdeutlicht.

Das Löschen von Kanten wird weiterhin dadurch motiviert, dass die Delaunay-Triangulation oftmals an den Bildrändern Knoten verbindet, die sehr weit voneinander entfernt sind. Dadurch entstehen lange Kanten, die das Phase Unwrapping erschweren. Aus diesem Grund sollten sie entfernt werden.

Eine weitere Forderung soll das Einfügen einer Kante nur erlauben, wenn sie eine Masche in zwei Teilflächen unterteilt. Dies entspricht dem Fall in Abbildung 4.12c. Auch diese Bedingung soll sicherstellen, dass jede Kante zu mindestens einer Masche gehört, und somit eindeutige, überlappungsfreie Vermaschungen in den Ebenen der Räumlichen Struktur vorliegen.

Im Folgenden werden nun die Klassen für die Räumliche Struktur vorgestellt. Dies geschieht wiederum anhand von UML-Diagrammen, mit deren Hilfe die Attribute und Methoden erläutert werden sollen.

Knotenliste_raum

Die Knoten in der Räumlichen Struktur werden in der Klasse *Knotenliste_raum* verwaltet, deren Modell dem UML-Diagramm in Abbildung 4.13 entspricht.



Abbildung 4.13: UML-Diagramm der Klasse *Knotenliste_raum*.

Das wichtigste Attribut dieser Klasse ist, wie aus dem Namen hervorgeht, eine Liste, in der die Objekte der Klasse *Knoten_raum* enthalten sind.

Ergänzt wird dies durch einige Verwaltungsstrukturen. Die erste enthält für jede Pixelkoordinate, die in der Datenstruktur enthalten ist, eine Liste mit Referenzen auf die Knoten, die diese Koordinaten gemeinsam haben. Sie wird beispielsweise dafür verwendet, die Delaunay-Triangulation von einer Ebene in die anderen

zu übertragen.

In der zweiten Struktur ist für jedes Bild des Stapels eine Liste mit den IDs der Knoten abgelegt, die in der zugehörigen Ebene der Knotenwolke liegen. Dieses Attribut wird benötigt, wenn eine Kante aus der zeitlichen Struktur gelöscht wird. Dies kommt dem Entfernen eines Interferogramms aus dem Bildstapel gleich. Entsprechend sind alle Knoten aus der Ebene der Räumlichen Struktur zu löschen, die mit diesem D-InSAR-Bild korrespondiert.

In der letzten Struktur ist für jeden Knoten eine Liste von IDs gespeichert, die auf die Kanten verweisen, die an diesem Knoten beginnen bzw. enden. Wie bereits in der Klasse *Knotenliste_zeit* wird diese für Nachbarschaftsanfragen an den Knoten verwendet.

liste ... Liste mit den Knotenobjekten der Räumlichen Struktur.
lagegleicheKnoten ... Referenzen auf Knoten, die gleiche Pixelkoordinaten haben.
bildgleicheKnoten ... Referenzen auf Knoten, die im gleichen Bild liegen.
ausgehendeKanten ... Referenzen auf die ausgehenden (bzw. ankommenden) Kanten für jeden Knoten.

Die erste Methode, die für diese Klasse vorgeschlagen wird, ist *gebeInteressanteKnoten()*. Mit ihr soll es möglich sein, nach dem Einlesen der Bildpunkte zu untersuchen, wie sich die Anzahl der Knoten pro Bild verändert, wenn die Schwellwerte der Auswahlkriterien nach oben korrigiert werden. Diese Methode soll Voruntersuchungen des Bildstapels ermöglichen, bevor eine Vermaschung der Daten in den Bildern durchgeführt wird.

Die Methode *gebeLagegleicheKnoten()* gibt Referenzen auf alle Knoten zurück, die die gleichen Pixelkoordinaten wie ein übergebener Knoten haben. Hier wird also auf eine Liste aus der oben genannten Verwaltungsstruktur zugegriffen. Eine ähnliche Aufgabe erfüllt die Methode *gebeBildgleicheKnoten()*, die entsprechend Referenzen auf alle Knoten in einem Bild zurückgibt.

projiziereInEbene() ist eine Methode, die für die spätere Delaunay-Triangulation benötigt wird. Sie gibt alle auftretenden Bildkoordinaten zurück sowie pro Koordinatenpaar eine Referenz auf die entsprechende Liste aus der Verwaltungsstruktur *lagegleicheKnoten*.

Die beiden Methoden *showInfo()* und *saveInfo()* dienen, wie in den anderen Listenklassen, der geordneten Ausgabe der Informationen zu den Knotenobjekten. Diese umfassen die Attribute der Klasse *Knoten_raum* sowie die Liste der IDs der ausgehenden Kanten.

Knoten_raum

In der Räumlichen Struktur werden die Knoten durch die Klasse *Knoten_raum* repräsentiert, deren Modell dem UML-Diagramm in Abbildung 4.14 entspricht.

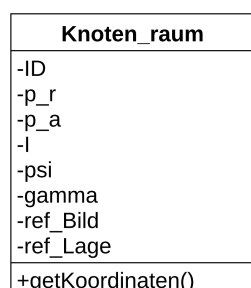


Abbildung 4.14: UML-Diagramm der Klasse *Knoten_raum*.

Auch für die Knoten der Räumlichen Struktur ist ein eindeutiger Bezeichner als Attribut vorgesehen, über

den das Objekt identifiziert und auf dieses zugegriffen werden kann.

Jeder Knoten entspricht einem Bildpunkt in drei Bildern, die sich jedoch alle dem gleichen differentiellen Interferogramm und somit auch der gleichen Kante in der Zeitlichen Struktur zuordnen lassen. Denn wie schon bei der Klasse *Kante_zeit* beschrieben, gehört zu jeder D-InSAR-Szene auch ein Intensitäts- und Kohärenzbild. Da diese jedoch alle auf dem gleichen Gitter vorliegen, kann jedem Knoten je eine Pixelposition in Range- und Azimuthrichtung zugewiesen werden. Entsprechend sind auch die Werte an diesem Bildpunkt als Attribut vorgesehen. Das wichtigste Attribut stellt der mehrdeutige Phasenwert $\psi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ dar, auf dem die Auswertung von differentiellen SAR-Interferogrammen gründet. Der Kohärenzwert $\gamma_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ ist ein qualitätsbestimmendes Merkmal (vgl. Abschnitt 2.4) und dient auch der Auswahl der Bildpunkte, wie zu Beginn dieses Abschnitts beschrieben wurde. Der Intensitätswert $I_{x(i,j)}^{\Delta t_{\alpha,\beta}}$ wird bisher nicht weiter benötigt, soll der Vollständigkeit halber allerdings auch in der Datenstruktur als Information enthalten sein.

Zusätzlich sollen zwei weitere Attribute definiert werden, die der Verknüpfung der Daten dienen. Das erste ist eine Referenz auf die Kante der Zeitlichen Struktur, die die eben genannten Bilder repräsentiert. Somit ist diese Referenz die Umsetzung der Assoziation zwischen der Klasse *Knote_raum* und *Kante_zeit* im UML-Diagramm in Abbildung 4.1. Diese Referenz soll zeitgleich auf die Liste der Verwaltungsstruktur *bildgleicheKnoten* der Klasse *Knotenliste_raum* verweisen, die der Ebene zugeordnet ist, in der der Knoten liegt. Somit ist es möglich, ausgehend von einem Knoten auf alle Knoten im gleichen Bild zuzugreifen.

Die zweite Referenz zeigt auf die Liste des Attributs *lagegleicheKnoten* der Klasse *Knotenliste_raum*, die den Koordinaten dieses Knoten zugewiesen ist.

Die eben definierten Eigenschaften können den Attributen im UML-Diagramm, wie folgt, zugeordnet werden:

ID	...	Die eindeutige ID für die Identifikation und den Zugriff auf den Knoten.
p_r	...	Pixelkoordinate in Rangerichtung.
p_a	...	Pixelkoordinate in Azimuthrichtung.
I	...	Intensitätswert $I_{x(i,j)}^{\Delta t_{\alpha,\beta}}$.
psi	...	Mehrdeutiger Phasenwert $\psi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$.
gamma	...	Kohärenzwert $\gamma_{x(i,j)}^{\Delta t_{\alpha,\beta}}$.
ref_Bild	...	Referenz auf die Kante in der Zeitlichen Struktur, die dem D-InSAR-Bild entspricht, und auf die Referenzstruktur, die alle Knoten in einem Bild verknüpft.
ref_Lage	...	Referenz auf die Referenzstruktur, die alle lagegleichen Knoten verknüpft.

Als Methoden werden hier die wieder die Standard Getter und Setter vorgeschlagen, sowie eine Methode *getKoordinaten()* zum Abrufen der Lagekoordinaten.

Kantenliste_raum

Die zweite Verwaltungsklasse der Räumlichen Struktur ist *Kantenliste_raum*, die über das UML-Diagramm in Abbildung 4.15 modelliert wird.

Das zentrale Attribut dieser Klasse ist eine Liste, die hier die Instanzen der Klasse *Kante_raum* enthält. Dazu sind erneut Verwaltungsstrukturen als Attribute vorgesehen.

Es soll eine Liste erzeugt werden, in der wiederum Listen enthalten sind, die auf lageidentische Kanten verweisen. Zwei Kanten sind lagegleich, wenn ihre Anfangs- und Endknoten jeweils identische Bildkoordinaten besitzen. Diese Struktur ist für spätere Anwendungen des EMCF-Algorithmus hilfreich.

Lagegleiche Kanten können den Kanten des Basisliniennetzwerks zugeordnet werden, sodass mit ihnen das Temporal Unwrapping durchgeführt werden kann. Des Weiteren wird dieses Attribut für das Löschen und Einfügen von Kanten benötigt. Wie bereits in der Einführung dieses Abschnitts erläutert, soll das Ändern

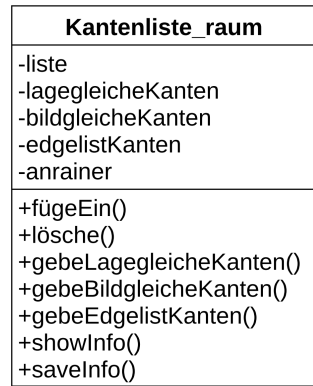


Abbildung 4.15: UML-Diagramm der Klasse *Kantenliste_raum*.

der Vermaschung in einer Ebene der Räumlichen Struktur die gleiche Modifikation in den anderen Ebenen zur Folge haben.

Die zweite Verwaltungsstruktur enthält Listen mit Kanten-IDs, die auf Kantenobjekte im gleichen Bild verweisen. Durch dieses Attribut wird das Löschen von D-InSAR-Szenen aus dem Bildstapel erleichtert. Sobald ein Interferogramm entfernt wird, müssen auch alle Kanten in der zugehörigen Ebene der Räumlichen Struktur eliminiert werden.

Abschließend soll es die dritte Struktur ermöglichen, auf alle Kanten zuzugreifen, die zwischen lagegleichen Knoten in benachbarten Ebenen der Knotenwolke aufgespannt wurden. Damit wird der Zugriff auf Kanten möglich, die für den Edgelist-Algorithmus benötigt werden und die noch nicht über die anderen Referenzstrukturen zugänglich sind. Außerdem wird dieses Attribut beim Einfügen und Entfernen von Szenen aus dem Bildstapel benötigt. Dies gleicht dem Einfügen bzw. Entfernen von Ebenen in der Räumlichen Struktur. Entsprechend sind Kanten zwischen diesen Ebenen an die neu entstehenden Nachbarschaften anzupassen. Auch für die Kanten der Räumlichen Struktur soll pro Kante eine Referenz auf den linken und rechten Anrainer gespeichert werden, wie es bereits in der Klasse *Kantenliste_zeit* erläutert wurde. Dies gilt jedoch nur für die Kanten innerhalb der Ebenen, da zwischen den Ebenen keine Flächen modelliert werden.

- liste** ... Liste mit den Kantenobjekten der Räumlichen Struktur.
lagegleicheKanten ... Referenzen auf Kanten, die Knoten mit gleichen Pixelkoordinaten haben.
bildgleicheKanten ... Referenzen auf Kanten, die im gleichen Bild liegen.
edgelistKanten ... Referenzen auf Kanten, die zwischen benachbarten Bildern aufgespannt wurden.
anrainer ... Referenzen auf die an die Kante angrenzenden Flächen.

Die Methoden *fügeEin()* und *lösche()* dienen ihrem Namen entsprechend dem Einfügen und Löschen von Kanten aus der Liste und werden für die Nachbearbeitung der Delaunay-Triangulation benötigt. Auch das Anpassen der Kanten zwischen den Ebenen soll durch diese Funktionen ermöglicht werden.

Die drei Methoden *gebeLagegleicheKanten()*, *gebeBildgleicheKanten()* und *gebeEdgelistKanten()* dienen dem lesenden Zugriff auf die entsprechenden Verwaltungsstrukturen dieser Klasse.

Wie bei den bisherigen Listenklassen können mit den Methoden *showInfo()* und *saveInfo()* die Informationen der Instanzen der Klasse *Kante_zeit* geordnet ausgegeben werden. Die Informationen beinhalten dabei Attribute des Kantenobjekts sowie die IDs der Anrainer.

Kante_raum

In Abbildung 4.16 ist das UML-Diagramm der Klasse *Kante_raum* dargestellt, welche die Kanten des Räumlichen Graphen modelliert.



Abbildung 4.16: UML-Diagramm der Klasse *Kante_raum*.

Die Kanten der Räumlichen Struktur sollen ebenfalls über eine spezifische ID identifiziert werden können, sodass auch ein Zugriff auf das Objekt möglich ist. Analog zu den Kanten des Basisliniennetzwerks wird der Bezug zum Anfangs- und Endknoten der Kante über zwei Attribute realisiert, die deren IDs enthalten. Die wichtigste Eigenschaft der Klasse *Kante_raum* ist der mehrdeutige Phasengradient $\psi_{\Delta x(i,j)}^{\Delta t_{\alpha,\beta}}$. Weiterhin sind einige Referenzen als Attribute vorgesehen, die zur Verknüpfung der Daten dienen. Die erste zeigt zum einen auf die mit dem D-InSAR-Bild korrespondierende Kante der Zeitlichen Struktur und zum anderen auf die zugehörige Liste der Verwaltungsstruktur *bildgleicheKanten* der Klasse *Kantenliste_raum*. Über die zweite Referenz soll es möglich sein ausgehend von einer Kante, auf alle Kanten zuzugreifen, die Knoten mit lagegleichen Koordinaten besitzen. Dazu verweist die Referenz auf das Attribut *lagegleicheKanten* der Klasse *Kantenliste_raum*.

Diese beiden Referenzen sollen nur für Kanten gesetzt werden, die in einem D-InSAR Bild liegen. Für Kanten, die zwischen lagegleichen Knoten in benachbarten Ebenen der Räumlichen Struktur aufgespannt werden, ist die letzte Referenz vorgesehen. Sie zeigt auf die entsprechende Liste der Struktur *edgelistKanten* der Klasse *Kantenliste_raum*.

- ID** ... Die eindeutige ID für die Identifikation und den Zugriff auf die Kante.
- ID_Anfang** ... ID des Startknotens der Kante.
- ID_Ende** ... ID des Endknotens der Kante.
- delta_psi** ... Mehrdeutiger Phasengradient $\psi_{\Delta x(i,j)}^{\Delta t_{\alpha,\beta}}$.
- ref_Bild** ... Referenz auf die Kante in der Zeitlichen Struktur, die dem D-InSAR-Bild entspricht, und auf die Referenzstruktur, die alle Kanten in einem Bild verknüpft.
- ref_Lage** ... Referenz auf die Referenzstruktur, die alle lagegleichen Kanten verknüpft.
- ref_Edgelist** ... Referenz auf die Referenzstruktur, die alle Kanten verknüpft, die zwischen lagegleichen Knoten in benachbarten Bildern aufgespannt wurden.

Abermals sind für die Klasse die Standard Getter- und Setter-Methoden vorgesehen, über die auf die Attribute lesend und schreibend zugegriffen werden kann. In Hinblick auf die Nutzung der Datenstruktur für den Edgelist-Algorithmus sollte zusätzlich eine Methode *getKantenrestriktion()* implementiert werden, welche die Restriktionen nach Gleichung (3.20) und (3.21) zurückgibt.

Flächenliste_raum

Flächen in der Räumlichen Struktur werden in der Klasse *Flächenliste_raum* gespeichert, deren UML-Diagramm in Abbildung 4.17 zu sehen ist.

Diese Klasse besitzt zwei Attribute. Zum einen eine Liste, die alle Flächenobjekte der Räumlichen Struktur enthält und zum anderen eine Verwaltungsstruktur. Diese enthält für jedes Bild eine Liste mit Referenzen auf die Flächen, die in eben diesem Bild liegen. Damit soll später das sogenannte Spatial Unwrapping erleichtert

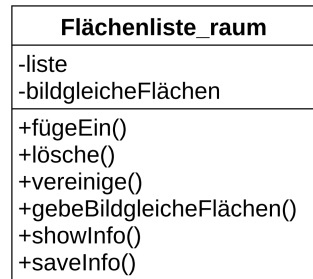


Abbildung 4.17: UML-Diagramm der Klasse *Flächenliste_raum*.

werden, das auf jedem Bild separat ausgeführt wird.

Außerdem wird dieses Attribut benötigt, um alle Flächen in einer Ebene der Räumlichen Struktur zu löschen, falls das zugehörige Interferogramm aus dem Bildstapel entfernt wird.

liste ... Liste mit den Flächenobjekten der Räumlichen Struktur.
bildgleicheFlächen ... Referenzen auf Flächen, die im gleichen Bild liegen.

Das Löschen und Einfügen von Flächen aus der bzw. in die Liste wird durch *lösche()* und *fügeEin()* ermöglicht. Die Notwendigkeit dieser Methoden wurde bereits in der Klasse *Flächenliste_zeit* erläutert. Gleiches gilt für die Methode *vereinige()*, mit der zwei Teilflächen zu einer gemeinsamen Masche zusammengefügt werden können.

gebeBildgleicheFlächen() dient dem Zugriff auf die entsprechende Verwaltungsstruktur.

Abschließend sind mit *showInfo()* und *saveInfo()* erneut zwei Methoden zur geordneten Ausgabe der Attribute der Räumlichen Flächenobjekte vorgesehen.

Fläche_raum

Abgeschlossen wird die Vorstellung der Modellierung der Räumlichen Struktur mit der Klasse *Fläche_raum*. Das dazugehörige UML-Diagramm ist in Abbildung 4.18 dargestellt.

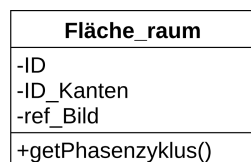


Abbildung 4.18: UML-Diagramm der Klasse *Fläche_raum*.

Die Klasse *Fläche_raum* besitzt wie alle Graphenobjekte einen eindeutigen Bezeichner zur Identifikation und für den Zugriff auf das Objekt. Analog zur Fläche in der Zeitlichen Struktur wird eine Liste der IDs der Kanten gespeichert, die die Fläche begrenzen. Auch hier muss auf die Kantenrichtungen geachtet werden. Zwar werden bei einer Delaunay-Triangulation automatisch positiv orientierte Flächen erzeugt, jedoch wird eine Kante in entgegengesetzten Richtungen traversiert, wenn man den positiven Zyklus der angrenzenden Flächen durchläuft. Daher wird bei der Delaunay-Triangulation zumeist eine Punktliste zur Darstellung der Dreiecke vorgezogen.

Hier werden jedoch Kanten zur Beschreibung einer Masche präferiert, da somit Phasenzyklen besser dargestellt werden können.

Abschließend besitzt diese Klasse eine Referenz auf die Kante der Zeitlichen Struktur, die mit dem

D-InSAR-Bild assoziiert ist, indem die Fläche liegt. Diese Referenz zeigt wie bei den Klassen *Knoten_raum* und *Kante_raum* auf eine Verwaltungsstruktur. Über sie ist es in diesem Fall möglich, auf alle Flächen zuzugreifen, die in einem Bild liegen.

- ID** ... Die eindeutige ID für die Identifikation und den Zugriff auf die Fläche.
ID_Kanten ... Liste der IDs der Kanten, die die Fläche begrenzen; mit Vorzeichen für die Orientierung.
ref_Bild ... Referenz auf die Kante in der Zeitlichen Struktur, die dem D-InSAR-Bild entspricht, und auf die Referenzstruktur, die alle Flächen in einem Bild verknüpft.

Die Klasse sollte ebenfalls die Standard Getter- und Setter-Methoden besitzen. Außerdem sollte eine Methode *getPhasenzyklen()* implementiert werden, mit der die Phasenzyklen berechnet werden, die als Restriktionen für das Phase Unwrapping benötigt werden (vgl. Gleichung (3.5)).

4.3 Anfragen an die Datenstrukturen

Bereits in Abschnitt 4.2.1 wurde herausgestellt, dass die Klasse *Bildstapel* als Nutzerschnittstelle zur modellierten Datenstruktur dient. Im Folgenden werden mögliche Anfragen vorgestellt, die über diese Klasse an die Datenstruktur gestellt werden können. Zur Übersicht wird dafür ein UML-Diagramm dieser Klasse in Abbildung 4.19 präsentiert.



Abbildung 4.19: UML-Diagramm der Klasse *Bildstapel*.

Die Attribute der Klasse *Bildstapel* sind Instanzen der in Abschnitt 4.2 vorgestellten Klassen der Räumlichen und Zeitlichen Struktur. Der Konstruktor dieser Klasse sollte überladen sein, damit eine Bearbeitung in verschiedenen Zuständen des Aufbaus möglich ist. Es werden drei Konstruktoren vorgeschlagen. Der erste soll lediglich die Zeitliche Struktur aufbauen, sodass diese individuell durch das Einfügen und Löschen von Kanten bearbeitet werden kann. Beim zweiten Konstruktor sollte zusätzlich die Knotenliste der Räumlichen Struktur erzeugt werden. Somit ist es möglich, die Auswahl der Knoten und deren Verteilung in den Bildern genauer zu untersuchen. Der letzte Konstruktor sollte sowohl die Zeitliche als auch die Räumliche Struktur erzeugen.

Im Folgenden werden die Methoden der Klasse vorgestellt, die die Anfragen an die Datenstruktur definieren.

leseKnotenEin():

Mit dieser Methode kann die Knotenliste der Räumlichen Struktur aufgebaut werden. Vorausgesetzt, vorher wurde lediglich die Zeitliche Struktur erzeugt. Auf diese Weise kann die Knotenauswahl anhand aller Bilder getroffen wird, die zu diesem Zeitpunkt im Bildstapel sind.

Wird die Zeitliche Struktur hingegen bei eingelesenen Knoten verändert, so entspricht dies dem Einfügen oder Entfernen eines Interferogramms in den bzw. aus dem Bildstapel. Entsprechend müssen die Knoten, die in dieser Aufnahme liegen, ebenfalls eingelesen bzw. gelöscht werden. Die Auswahl der interessanten Punkte anhand des Kohärenzschwellwerts und des prozentualen Anteils (vgl. Abschnitt 3.3) bezieht sich jedoch weiterhin auf den ursprünglichen Bildstapel.

vermasche():

Diese Methode erzeugt die Kanten- und Flächenliste der Räumlichen Struktur. Dies ist jedoch die zeitintensivste Operation, weshalb es sinnvoll sein kann, die Listen erst zu einem späteren Zeitpunkt zu erzeugen.

scatter():

Die Scatter-Methode dient der Darstellung der Knotenverteilung der Räumlichen Struktur in den Bildern. Außerdem sollten auch die Knoten markiert werden, die den festgelegten Kohärenzschwellwert überschreiten. Somit wird ein Eindruck über die Qualität des Interferogramms vermittelt, dessen Knoten gerade visualisiert werden. Ebenfalls soll es mit der Methode realisierbar sein, den Effekt von restriktiveren Parametern bei der Knotenauswahl zu visualisieren.

plot():

Die Plot-Methode soll die Bearbeitung der Datenstruktur interaktiv unterstützen. Es soll möglich sein, wahlweise die Zeitliche oder Räumliche Graphenstruktur darzustellen. Dabei kann die Vermaschung der Räumlichen Knoten nur für ein individuelle Ebene angezeigt werden.

Durch Anklicken eines Elements können Informationen zu dem zugehörigen Objekt aus diesem Plot abgerufen werden. So sollte für einen Knoten mindestens dessen ID und eventuell dessen Koordinaten angezeigt werden. Für eine Kante sollte ebenfalls deren ID und zusätzlich die ID des Anfangs- und Endknoten sowie die IDs der Anrainer dargestellt werden. Durch die IDs der Anfangs- und Endknoten ist es möglich, die Richtung der Kante zu bestimmen. Die IDs der Anrainer gestatten die Identifikation der Maschen.

showInfo():

Die Methode *showInfo()* erzeugt eine geordnete Konsolenausgabe der Informationen zu einem bestimmten Objekt, das über seine ID identifiziert wird. Diese Methode wurde für jede Listenklasse in Abschnitt 4.2.2 und 4.2.3 genauer erläutert. Die Funktion der Klasse *Bildstapel* soll die gleichnamigen klassenspezifischen Methoden aufrufen.

saveInfo():

Diese Funktion soll ebenfalls die gleichnamigen klassenspezifischen Methoden aufrufen, die für die Listenklassen der Räumlichen und Zeitlichen Struktur definiert wurden. Sie erzeugt eine Textdatei mit den Informationen zu den Objekten, die in den Listen gespeichert sind.

löscheKante():

Mit dieser Methode können Kanten aus der Zeitlichen oder Räumlichen Struktur gelöscht werden. Dabei ist es notwendig, nicht nur das Objekt aus der entsprechenden Liste zu löschen, sondern auch alle klassenübergreifenden Konsequenzen zu behandeln. Dazu gehören die Vereinigung zweier Flächen und das Entfernen von Flächen aus der Datenstruktur, wie in den Abschnitten 4.2.2 und 4.2.3 beschrieben wurde. Außerdem

müssen für die Knoten die Liste der ausgehenden Kanten und für die Kanten die Liste der Anrainer aktualisiert werden.

Wird eine Kante aus der Zeitlichen Struktur gelöscht, hat dies ebenfalls Auswirkungen auf die Räumliche Struktur, sofern diese bereits aufgebaut wurde. Diese Operation entspricht dem Entfernen eines D-InSAR-Bildes aus dem Stapel. Die Konsequenzen wurden an den entsprechenden Stellen in den Abschnitten 4.2.2 und 4.2.3 diskutiert.

fügeKanteEin():

Das Einfügen von Kanten in die Räumliche oder Zeitliche Struktur umfasst nicht nur das Einfügen des neuen Objekts in die entsprechende Liste. Auch hier müssen mehrere Folgen berücksichtigt werden. Einerseits sollen neue Flächen erzeugt werden, wenn durch die Kante eine Masche geteilt wird. Weiterhin müssen auch hier die Listen mit den ausgehenden Kanten für die Knoten aktualisiert werden, zwischen denen die Kante eingefügt wird. Außerdem muss die Liste der Anrainer angepasst werden, sobald neue Flächen entstehen.

Das Einfügen einer Kante in die Zeitliche Struktur hat zudem Auswirkungen auf die Räumliche Struktur. Folglich ist es notwendig, Knoten, Kanten und Flächen in dem Bild zu erzeugen, das mit der eingefügten Kante korrespondiert. Dies ist allerdings nur erforderlich, wenn die entsprechenden Strukturen bereits aufgebaut wurden. Außerdem müssen auch die Kanten, die zwischen benachbarten Ebenen in der Räumlichen Struktur aufgespannt wurden, an der entsprechenden Stelle angepasst werden.

getPhasenzyklen():

Diese Methode dient der Abfrage der Informationen, die für das Temporal bzw. das Spatial Unwrapping des EMCF-Algorithmus benötigt werden.

Wie im Abschnitt 3.3 zu diesem Algorithmus erläutert wurde, kann jede Menge von lagegleichen Kanten den Kanten des Basisliniennetzwerkes zugeordnet werden. Für das Temporal Unwrapping sollten also folgende Informationen zurückgegeben werden. Zum einen wird für jede Fläche der Zeitlichen Struktur die Liste mit den IDs der Kanten benötigt, die den Zyklus bilden. Diese Ausgabe muss um eine Liste ergänzt werden, die für jede dieser Kanten die relativen Basislinien enthält und den Phasengradienten aus der Menge der lagegleichen Kanten, der der Kante der Zeitlichen Struktur zugeordnet werden kann.

Für das Spatial Unwrapping sollten für alle Flächen in einem Bild die Liste der Kanten-IDs zurückgegeben werden. Zusätzlich sollte auch der Wert der Restriktion für jede Fläche beigefügt werden, der sich aus den Phasengradienten der Kanten bestimmen lässt (vgl. Gleichung (3.5)).

getEdgelistRestriktionen():

Wie der Name dieser Methode bereits andeutet, soll diese Funktion die Restriktionen zurückgeben, die für das Phase Unwrapping mit dem Edgelist-Algorithmus benötigt werden. Dabei sollen allerdings nicht nur für jede Kante der Räumlichen Struktur die Restriktionen nach Gleichung (3.20) und (3.21) berechnet werden. Die Ausgabe sollte zusätzlich die Kanten-IDs sowie die zugehörigen IDs der Anfangs- und Endknoten umfassen.

5. Implementierung

Im vorangegangenen Kapitel 4 wurde ein Konzept für eine Datenstruktur vorgestellt, mit der die raum/zeitlichen Zusammenhänge in einem D-InSAR-Bildstapel dargestellt werden können. Dieses theoretische Konzept soll durch eine Implementierung ergänzt werden, mit der die Umsetzbarkeit und Effizienz nachgewiesen werden soll.

Wie schon bei der Modellierung wurde hierfür ein objektorientierter Ansatz verfolgt. Als Programmiersprache bzw. -umgebung wurde *MATLAB* gewählt. Die ist dadurch begründet, dass in der Professur für Theoretische Geodäsie der Universität Bonn durch WALZOG (2014) bereits eine Implementierung des Edgelist-Algorithmus (vgl. Abschnitt 3.4) in *MATLAB* vorhanden ist. Außerdem befindet sich eine Umsetzung des EMCF-Algorithmus (vgl. Abschnitt 3.3) bereits im Aufbau. Durch die Programmierung von Schnittstellen zu diesen Programmen soll die Datenstruktur zu einem späteren Zeitpunkt praktische Anwendung bei der Auswertung von D-InSAR-Bildstapeln finden.

Zu diesem Zeitpunkt ist die Umsetzung in weiten Teilen abgeschlossen. Deren Lauffähigkeit soll unter Verwendung von Realdaten nachgewiesen werden. Daher wird im folgenden Abschnitt 5.3 die Auswahl eines Testdatensatzes aus der Datengrundlage präsentiert. Mit diesen Daten wurde die Lauffähigkeit und Effizienz der Implementierung anhand eines Testskripts überprüft. Dies wird in Form eines Funktionshandbuchs in Anhang C dokumentiert. In Abschnitt 5.2 ist eine Funktionsübersicht der implementierten und getesteten Funktionen samt Kurzbeschreibungen zu finden.

5.1 Auswahl eines Testdatensatzes

Im folgenden Abschnitt soll der Testdatensatz vorgestellt werden, mit dem die Lauffähigkeit der Implementierung der Datenstruktur nachgewiesen werden soll. Dieser Testdurchlauf wird mit dem Funktionshandbuch in Anhang C dokumentiert. Zur Ergänzung des Handbuchs werden Abbildungen erstellt, mit denen die verschiedenen Operationen auf die Datenstruktur veranschaulicht werden sollen. Um diese übersichtlicher zu gestalten, wird aus der Datengrundlage lediglich ein Subset entnommen.

Tabelle 5.1: Beschreibung des Testdatensatzes, der aus der Datengrundlage entnommen wurde.

ID (Slave)	ID (Master)	$\Delta t_{\alpha,\beta}$ [d]	$\Delta b_{\perp \alpha,\beta}$ [m]	$\Delta f_{c \alpha,\beta}$ [Hz]	Aufnahmedatum (Slave)	Aufnahmedatum (Master)
35	32	105	76.535	35.094	7-JAN-1997	24-SEP-1996
75	32	175	-62.307	27.310	18-MAR-1997	24-SEP-1996
37	32	245	41.194	31.417	27-MAY-1997	24-SEP-1996
37	35	140	-35.317	3.677	27-MAY-1997	7-JAN-1997
37	75	70	103.542	4.107	27-MAY-1997	18-MAR-1997
38	35	175	131.747	46.217	1-JUL-1997	7-JAN-1997
38	37	35	167.077	49.894	1-JUL-1997	27-MAY-1997
39	75	140	-15.311	7.277	5-AUG-1997	18-MAR-1997
39	37	70	-118.882	11.384	5-AUG-1997	27-MAY-1997
41	75	210	-194.458	5.818	14-OCT-1997	18-MAR-1997
41	39	70	-179.135	1.459	14-OCT-1997	5-AUG-1997
48	37	455	-36.847	13.622	25-AUG-1998	27-MAY-1997
48	38	420	-203.927	36.272	25-AUG-1998	1-JUL-1997
48	39	385	82.038	25.006	25-AUG-1998	5-AUG-1997
48	41	315	261.241	23.547	25-AUG-1998	14-OCT-1997

Zur Erzeugung dieses Subsets werden als erstes aus dem gegebenen D-InSAR-Bildstapel 15 Aufnahmen gewählt. Dabei wird darauf geachtet, dass diese Auswahl eine Dreiecksvermaschung in der Basislinienebene bildet, da diese Bedingung momentan noch von der Implementierung vorausgesetzt wird. Aus der gegebenen Datei zur Beschreibung der Datengrundlage (vgl. Abschnitt 2.5.2) wird ein entsprechender Auszug erstellt. Dieser ist der Tabelle 5.1 zu entnehmen. Es muss noch angemerkt werden, dass in diesem Subset mindestens ein Interferogramm enthalten sein muss, das die Referenzszene (Aufnahmedatum: 18.03.1997) als Master oder Slave besitzt. Da die Referenzszene den einzigen Punkt in der $T \times B_{\perp}$ -Ebene darstellt, dessen absolute Lage mit $P(0,0)$ bekannt ist, wird dieser benötigt, um aus den relativen Angaben der Tabelle 5.1 die Koordinaten der anderen Knoten zu bestimmen.

Für die so gewählten Bilder wird außerdem noch ein räumliches Subset gebildet. Folglich wird nur ein Teil der Aufnahmen verwendet. Dafür wird eine 500×500 Pixel großer Ausschnitt gewählt, dessen Lage dem roten Kasten in Abbildung 5.1 entspricht. Die weiße Linie entspricht dem Rahmen aus Abbildung 2.5. Somit kann ein eindeutiger Lagebezug zu den Originalbildern hergestellt werden.

Einbezogen werden in dieses Subset das Stadtgebiet von Köln sowie die Regionen südlich und westlich von Köln. Somit umfasst dieses sowohl urbane als auch ländliche Areale. Aus diesem Grund kann mit unterschiedlichen Dekorrelationseffekten gerechnet werden und die Dünnbesetztheit der auswertbaren Knoten wird in den Bildern deutlich.



Abbildung 5.1: Lage des Testdatensatzes im Aufnahmegebiet. Das Bild wurde mit der NEST Toolbox Version 5.1 erzeugt (EUROPEAN SPACE AGENCY (ESA), 2015).

In Abbildung 5.2a ist der Graph des Testbildstapels in der Basislinienebene dargestellt. Für den Graphen der Räumlichen Struktur wird exemplarisch die Vermaschung im ersten Bild des Teststapels abgebildet, da diese in allen Interferogrammen identisch ist (vgl. Abbildung 5.2b). Dabei wurden die Knoten, die in die

Räumliche Struktur eingelesen wurden, so ausgewählt, dass sie einen Kohärenzschwellwert von 0,35 in mindestens 30% der Bilder überschreiten. An der Größe der Maschen ist auch zu erkennen, dass die kohärenten Pixel in den Interferogrammen dünnbesetzt sind. Während in den urbanen Regionen die Triangulation sehr kleinmaschig ist, werden die Dreiecke in den ländlichen Arealen größer. Hier können also nur noch wenige Knoten zur Auswertung genutzt werden.

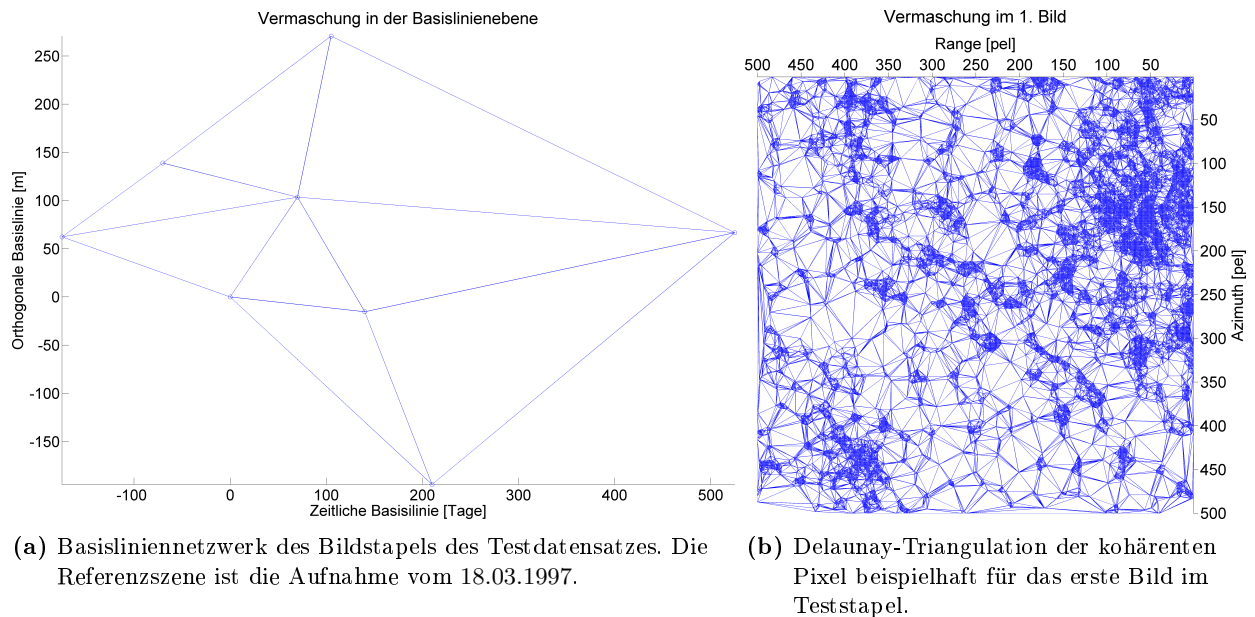
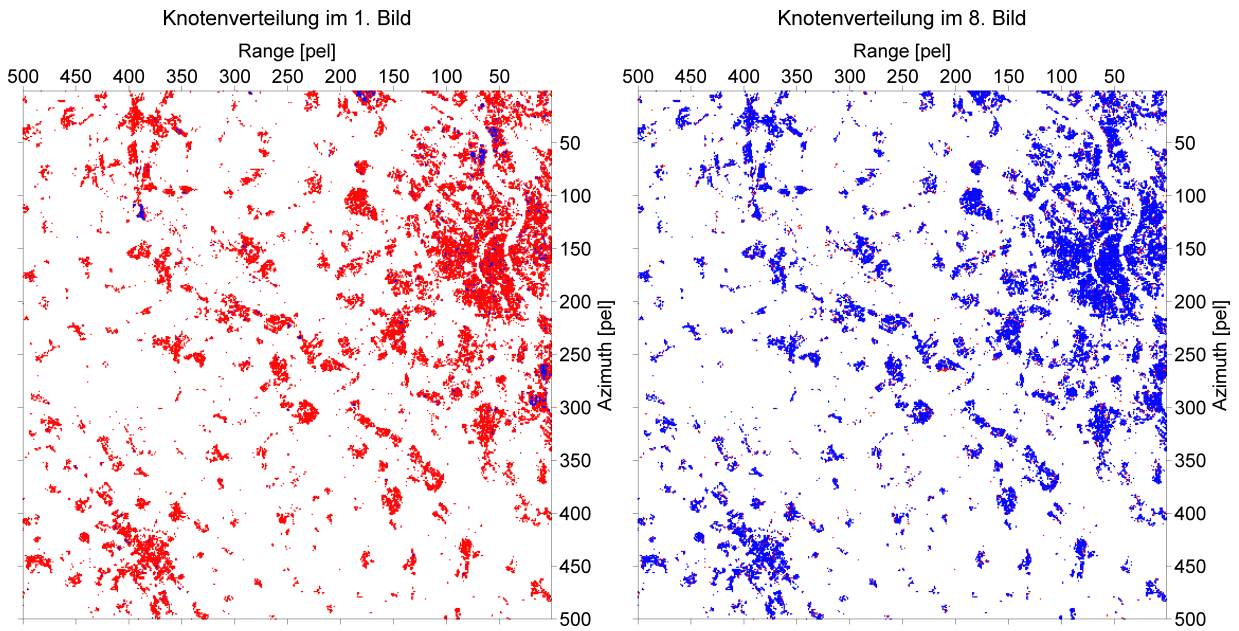


Abbildung 5.2: Zeitliches (links) und Räumliches (rechts) Netzwerk des Testdatensatzes.

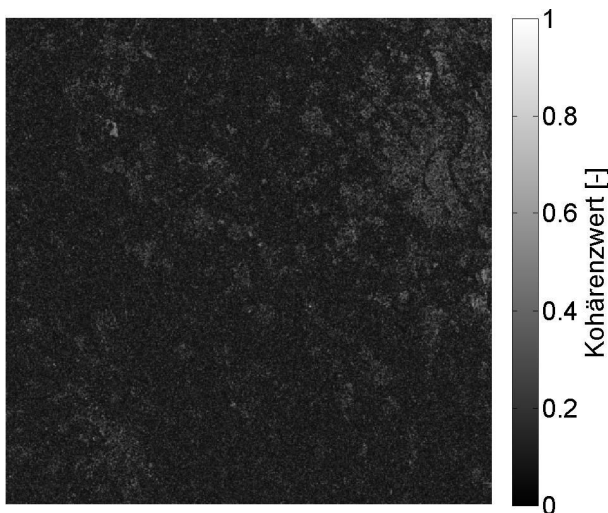
Mit der Abbildung 5.3 werden zwei Bilder aus dem Testdatensatz vorgestellt, die verdeutlichen, dass die hier aufgebaute Datenstruktur zur Untersuchung eines Bildstapels geeignet ist. In den Abbildungen 5.3a und 5.3b, ist zu oberst die Verteilung der ausgewählten Knoten im ersten und achten Interferogramm des Bildstapels aufgetragen. Dabei entsprechen die blauen Punkte den Knoten, die den Kohärenzschwellwert erfüllen, und die roten Punkte den Knoten, die den Schwellwert nicht erfüllen. Für das linke Bild kann daraus abgeleitet werden, dass seine weitere Auswertung problematisch werden könnte, da nur wenige blaue Punkte vorkommen. Dies wird durch die darunter abgebildeten, zugehörigen Datenprodukte untermauert. Das Kohärenzbild in Abbildung 5.3c weist ein sehr niedriges Kohärenzniveau auf, was sich durch starkes Phasenrauschen in Abbildung 5.3e bemerkbar macht. Für die rechten Abbildungen verhält es sich genau umgekehrt.

In Abschnitt 2.4 wurden die relativen Basislinien und die Doppler-Zentroid-Differenz als qualitätsbestimmende Merkmale für Interferogramme eingeführt. Entsprechend sollte zu erwarten sein, dass diese Werte für das linke Bild größer sind als für das rechte. Werden nun die den Bildern zugeordneten Werte in der ersten und achten Zeile der Tabelle 5.1 verglichen, zeigt sich, dass diese für beide Interferogramme relativ klein sind. Dies bedeutet, dass kleine relative Basislinien und eine kleine Doppler-Frequenz-Differenz nicht zwangsläufig eine gutes Interferogramm garantieren. Deswegen sollte durch eine optische Untersuchung die Nachbearbeitung eines Bildstapels unterstützt werden, was durch die Scatter-Plots der hier entwickelten Datenstruktur möglich wird.

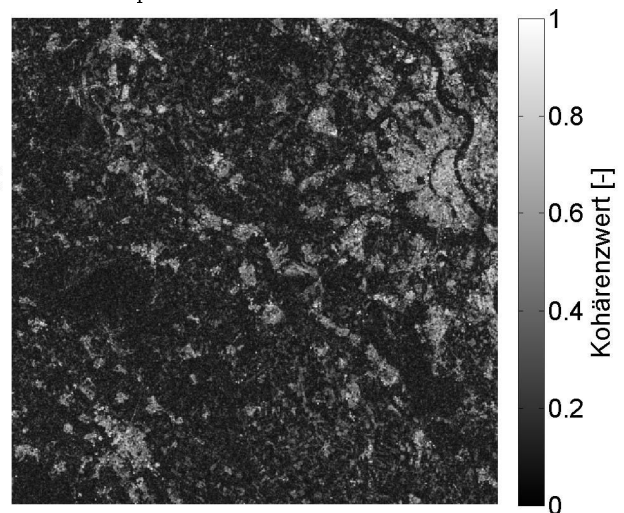


(a) Verteilung der ausgewählten Knoten der Räumlichen Struktur im ersten Bild des Teststapels.

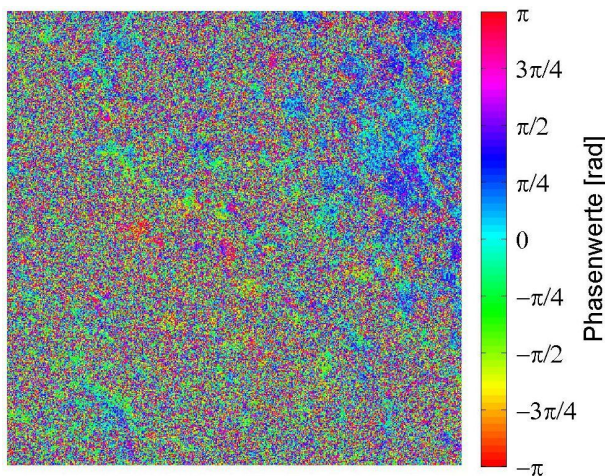
(b) Verteilung der ausgewählten Knoten der Räumlichen Struktur im achten Bild des Teststapels.



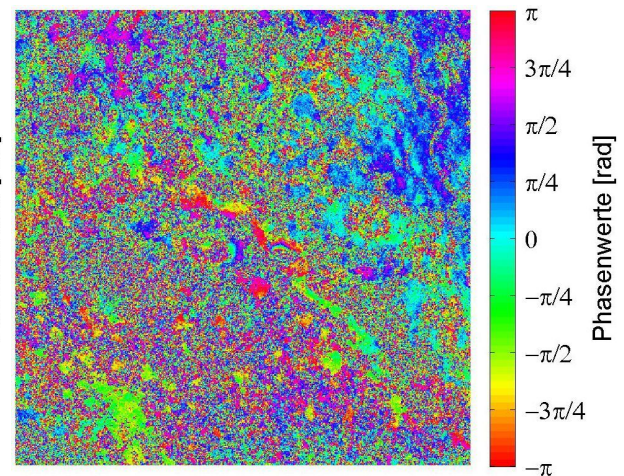
(c) Erstes Kohärenzbild im Teststapel.



(d) Achstes Kohärenzbild im Teststapel.



(e) Erstes D-InSAR-Bild im Teststapel.



(f) Achstes D-InSAR-Bild im Teststapel.

Abbildung 5.3: Gegenüberstellung der Daten, die an erster und achter Stelle im Teststapel liegen. Diese sind in den entsprechenden Zeilen der Tabelle 5.1 beschrieben. Hier wird der Einfluss der Kohärenz auf die Testdaten und die Knotenauswahl verdeutlicht.

5.2 Funktionsübersicht

Dieser Abschnitt dient als Funktionsübersicht für die implementierten Methoden der in dieser Arbeit entwickelten Datenstruktur. Die Übersicht dient gleichzeitig als Verzeichnis für das Funktionshandbuch im Anhang C, in dem die Methoden ausführlicher beschrieben werden.

In Tabelle 5.2 sind dazu, neben dem Namen der Funktion, auch Kurzbeschreibungen der Methoden zu finden. Für die Navigation im Handbuch sind zusätzlich die Ordnungsnummer und die Seitenzahl aufgeführt.

Tabelle 5.2: Funktionsübersicht mit Kurzbeschreibung und Referenz auf das Funktionshandbuch.

Funktionsname	Kurzbeschreibung	Ordnungsnummer	Seitenzahl
Bildstapel()	Der Konstruktor erzeugt eine Referenz auf eine Instanz der Klasse <i>Bildstapel</i> .	C.1	v
leseKnotenEin()	Mit dieser Funktion werden die Knoten der Räumlichen Struktur für den aktuellen Bildstapel eingelesen.	C.2	vii
vermasche()	Diese Funktion erzeugt die Vermaschung der Räumlichen Struktur. In der aktuellen Programmversion werden nur die Räumlichen Kanten in den Interferogrammen erzeugt.	C.3	viii
scatter()	Mit dieser Funktion ist es möglich, einen Scatter-Plot der Knotenverteilung in einem Bild darzustellen. Dabei werden Knoten, die den Kohärenzschwellwert unter- bzw. überschreiten, unterschiedlich eingefärbt. Zusätzlich kann die Knotenauswahl für restriktivere Parameter geplottet werden.	C.4	ix
plot()	Diese Funktion erzeugt einen interaktiven Plot des Basisliniennetzwerkes oder der Vermaschung der Knoten in einem ausgewählten Bild. Durch Anklicken der Elementen mit dem <i>DataCursor</i> lassen sich die Informationen zum ausgewählten Element abrufen.	C.5	xi
showInfo()	Mit dieser Funktion kann eine geordnete Konsolenausgabe der Informationen eines beliebigen Knoten-, Kanten- oder Flächenobjekts aus der Datenstruktur erstellt werden.	C.6	xiv
saveInfo()	Dies Funktion schreibt die Informationen zu allen Objekten einer der Knoten-, Kanten- oder Flächenklassen aus der Datenstruktur geordnet in eine Textdatei.	C.7	xvi
loescheKante()	Mit dieser Funktion wird es möglich sein, Kanten aus der Zeitlichen und Räumlichen Struktur zu entfernen. In der aktuellen Programmversion ist dies allerdings nur für Kanten im Basisliniennetzwerk möglich.	C.8	xvii
fuegeKanteEin()	Diese Funktion soll das Einfügen von Kanten in die Räumliche und Zeitliche Struktur ermöglichen. In der aktuellen Version ist momentan nur das Einfügen in das Basisliniennetzwerk umgesetzt.	C.8	xxi

6. Schlussbemerkungen

In diesem abschließenden Kapitel werden in Abschnitt 6.1 die in dieser Arbeit vorgestellten Inhalte nochmals zusammengefasst. In Abschnitt 6.2 wird zur hier entwickelten Datenstruktur kritisch Stellung genommen und ein Ausblick auf weiterführende Aufgaben gegeben. Diese umfassen zum einen Optimierungen und zum anderen Erweiterungen der Datenstruktur.

6.1 Fazit

Ziel dieser Arbeit war es, eine Datenstruktur zu entwickeln, mit der räumliche und zeitliche Zusammenhänge zwischen kohärenten Pixeln in einem Bildstapel von differentiellen SAR-Interferogrammen dargestellt werden können.

Um ein Verständnis der Daten beim Leser zu schaffen, wurden in Kapitel 2 die Grundlagen aufgearbeitet, die zur Beschreibung eines D-InSAR-Bildstapels nötig sind. Dafür wurde in den Abschnitten 2.1 bis 2.3 zuerst erläutert, was unter einem D-InSAR-Bild zu verstehen ist, um danach in Abschnitt 2.4 auf die Definition kohärenter Pixel einzugehen. Außerdem wurde gezeigt, wie sich die Kohärenz auf die Phasen in den Interferogrammen auswirkt und welche Einschränkungen sich dadurch für die Auswertung ergeben. Die Auswirkungen der Auswahl von SAR-Aufnahmepaaren zur Erzeugung von D-InSAR-Bildern auf den Bildstapel wurden in Abschnitt 2.5 eingehender untersucht. Hierbei wurde deutlich, dass gerade die Wahl der Datenpaare von den Auswertelgorithmen abhängt, die auf den Stapel angewendet werden sollen. Dazu wurden die inneren Strukturen eines Bildstapels dargelegt und erläutert, wie sich diese als Graph in der Basislinienebene darstellen lassen. Abschließend erfolgte die Vorstellung der Datengrundlage dieser Arbeit in Abschnitt 2.5.2 sowie die Eingliederung in die Ausführungen vorheriger Abschnitte.

Im folgenden Kapitel 3 wurden einige Anwendungen präsentiert, die als weitere Prozessierungsschritte für einen D-InSAR-Bildstapel in Frage kommen. Maßgeblich war hierbei die Auswahl anhand der Datengrundlage. Es wurden also nur solche Algorithmen gewählt, die sich auf den gegebenen Datenstapel anwenden lassen. Neben der Auswahl der Datenpaare legen diese Prozessierungsschritte fest, welche Zusammenhänge in der Datenstruktur darzustellen sind und welche Anfragen unterstützt werden sollen. Um Deformationsinformationen aus einem D-InSAR-Bildstapel zu berechnen, ist es nötig, aus den mehrdeutigen Phasen der Interferogramme, die absoluten Phasenwerte zu bestimmen. Aus diesem Grund wurden in Abschnitt 3.1 die Grundlagen des Phase Unwrappings erläutert. Anschließend wurde in Abschnitt 3.2 das Phase Unwrapping als MCF-Problem dargestellt, da auf diesem Ansatz spezielle Methoden zur Phasenabwicklung basieren, die auf einen gesamten Bildstapel angewendet werden können. Als erste dieser speziellen Methoden wurde in Abschnitt 3.3 der EMCF-Algorithmus von PEPE UND LANARI (2006) genannt, der die Datenstruktur maßgeblich beeinflusst hat. Die zweite 3D-Phase Unwrapping-Strategie, der Edgelist-Algorithmus nach (SHANKER UND ZEBKER, 2010), wurde in Abschnitt 3.4 präsentiert. Abgeschlossen wurden die theoretischen Ausführungen mit der Darstellung der SBAS-Methode in Abschnitt 3.5, die es ermöglicht, aus einem Bildstapel von abgewickelten D-InSAR-Szenen Deformationen zu bestimmen.

Der praktische Teil dieser Arbeit bestand in der Entwicklung der Datenstruktur. Dazu wurde in Kapitel 4 eine mögliche Modellierung dieser Datenstruktur vorgeschlagen. Abschnitt 4.1 fasst hierfür die Anforderungen zusammen, die die vorgestellten Anwendungen an die Struktur stellen. Anschließend wurde in Abschnitt 4.2 das in dieser Arbeit entwickelte Modell ausführlich erläutert. Dieses umfasst zwei Substrukturen, die *Zeitliche* und *Räumliche Struktur* genannt wurden. Die erste soll das Basisliniennetzwerk darstellen, das primär die

zeitlichen Informationen des Bildstapels enthält. Die Räumliche Struktur stellt hingegen die Zusammenhänge zwischen den kohärenten Pixeln dar. Verwaltet werden diese Substrukturen von einer Bildstapel-Klasse, die als Nutzerschnittstelle dient. Letztere wurde in Abschnitt 4.3 vorgestellt. Dort wurden auch mögliche Anfragen und Methoden präsentiert, die von der Datenstruktur unterstützt werden.

Die tatsächliche Umsetzung des Modells wurde dann in Kapitel 5 beschrieben. Hier wurde zunächst auf die Wahl einer geeigneten Programmierumgebung zur Realisierung des Modells eingegangen, wobei sich ergeben hat, eine objektorientierte Umsetzung in Matlab anzustreben. Die Lauffähigkeit der Implementierung anhand von Realdaten wurde mit einem Testdatensatz nachgewiesen, welcher in Abschnitt 5.1 eingeführt worden ist. In Form eines Funktionshandbuchs in Anhang C wurde dieser Testlauf ausführlich dokumentiert, wobei eine Ergänzung durch eine Funktionsübersicht in Abschnitt 5.2 vorliegt.

6.2 Ausblick

Wie bereits in Abschnitt 4.1 herausgestellt wurde, ist das vorgeschlagene Modell für die Datenstruktur nicht abschließend. Dieses kann und soll nach Bedarf erweitert werden. Eine mögliche Erweiterung, die dabei von besonderem Interesse ist, stellt die Entwicklung von Schnittstellen zu bestehenden Implementierungen des EMCF- und Edgelist-Algorithmus dar. Aufgabe dieser Schnittstellen wäre es, aus den Anfragen, die in Abschnitt 4.3 vorgestellt wurden, die Eingabeparameter für die Phase Unwrapping-Methoden zu erzeugen. Hier kann es unter Umständen nötig sein, die in der Datenstruktur vorgesehenen Attribute zu ergänzen. Denkbar wären beispielsweise Attribute für Kantengewichte sowie für abgewinkelte Phasengradienten und -werte.

Das Ziel dieser Arbeit ist mit der Entwicklung der Datenstruktur in Kapitel 4 zwar erreicht, jedoch gibt es bei der Implementierung noch an einigen Aufgaben, die es umzusetzen gilt.

Es wäre beispielsweise wünschenswert Datenstapel abzubilden, die aus mehreren Subsets bestehen, wie sie vom EMCF-Algorithmus und der SBAS-Methode vorgesehen sind. Des Weiteren wird bisher davon ausgegangen, dass der eingelesene Datenstapel eine Dreiecksvermaschung in der Basislinienebene bildet. Eine Verallgemeinerung der Struktur ist derzeit nur durch Nachbearbeitung möglich. Es wäre allerdings wünschenswert, dass die Datenstruktur einen allgemeinen Datenstapel im Sinne der SBAS-Methode einlesen kann. Anschließend sollte im Graphen in der Basislinienebene eine überlappungsfreie Vermaschung gefunden werden können, die für den EMCF-Algorithmus verwendbar wäre. Derzeit erweist sich das Finden der Maschen als problematisch, die links und rechts an eine Kante angrenzen. Ausgehend von einer Kante entspricht dies dem Finden von zwei positiv orientierten, kleinsten Zyklen.

In Kapitel 5 wurde außerdem erwähnt, dass auch noch einige grundlegendere Aufgaben ausstehen. Dazu gehört die Umsetzung der Klassen *Fläche_raum* und *Flächenliste_raum*. Folglich müssen auch die Methoden zum Einfügen und Löschen von Kanten für die Räumliche Struktur implementiert werden, da diese die eben genannten Klassen voraussetzen.

Außerdem muss noch die Erzeugung und Verwaltung von Kanten implementiert werden, die zwischen benachbarten Ebenen der Räumlichen Datenstruktur aufgespannt werden. Diese werden für den Edgelist-Algorithmus benötigt.

7. Danksagung

Zunächst möchte ich mich bei Prof. Dr. techn. Wolf-Dieter Schuh bedanken. Zum einen dafür, dass er mir die Möglichkeit geboten hat, diese Arbeit an der Professur für Theoretische Geodäsie am Institut für Geodäsie und Geoinformation der Universität Bonn zu schreiben. Zum anderen für die Unterstützung und Anmerkungen, die er mir im Laufe der Arbeit hat zukommen lassen.

Weiterhin möchte ich mich recht herzlich bei meiner Betreuerin M.Sc. Christina Esch bedanken, die mit ihrer freundlichen und hilfsbereiten Art stets bei Problemen helfen konnte. Außerdem haben mir die angeregten Diskussionen mit ihr dabei geholfen, die Thematik dieser Arbeit aufzubereiten.

Bei DI Dr. Karlheinz Gutjahr vom Institut für Informations- und Kommunikationstechnologien (DIGITAL) der Forschungsgesellschaft mbH Joanneum Research in Graz, Österreich, möchte ich mich für die Bereitstellung der Datengrundlage für diese Arbeit bedanken.

Mein besonderer Dank gilt meiner Freundin B.Sc. Katharina Franz, nicht nur für die Geduld und Unterstützung während der Anfertigung dieser Arbeit, sondern auch für die Korrekturen dieser Arbeit, für die sie sich auch in der Schlussphase immer kurzfristig Zeit genommen hat.

Schließlich gebührt ein ganz besonders herzlicher Dank meiner Mutter Rita, die mich während meines gesamten Studiums unterstützt hat und stets Geduld mit mir hatte.

A. Bezeichnungen

Allgemeines

$\#$	Anzahl
M'	Anzahl der Interferogramme im Bildstapel
N'	Anzahl der SAR-Aufnahmen
M	Anzahl der Kanten einer Triangulation in der $A_Z \times R_G$ -Ebene
N	Anzahl der Flächen einer Triangulation in der $A_Z \times R_G$ -Ebene
$A_Z \times R_G$	von der Azimuth- und Rangerichtung aufgespannte Ebene
$T \times B_\perp$	von der zeitlichen und orthogonalen Basislinie aufgespannte Ebene

Zufallszahlen

\mathcal{A}	Zufallsvektor
$E\{\mathcal{A}\}$,	Erwartungswert eines Zufallsvektors
$\Sigma\{\mathcal{A}\}$	Kovarianzmatrix eines Zufallsvektors

Operatoren

$(\cdot)^*$	komplexe Konjugation
$\langle \cdot \rangle_{-\pi, \pi}$	2π -Modulo-Operator, der einen Phasenwert in den mehrdeutigen Wertebereich $[-\pi, \pi)$ überführt
$[\cdot]$	Rundungsoperator zum nächsten Integer

Matrizen und Vektoren

x, α	Skalar
\mathbf{a}	Vektor
\mathbf{A}	Matrix
$\mathbf{A}_{[m \times n]}$	Matrix der Dimension $m \times n$
$A(m, n)$	Eintrag einer Matrix an an der Position (m, n)
\mathbf{A}^{-1}	Inverse einer Matrix
\mathbf{A}^-	Pseudoinverse einer Matrix

Verwendete Formelsymbole

Interferometrische Phase

t_α	Aufnahmezeitpunkt der α -ten SAR-Szene
\mathbf{t}	chronologisch sortierter Vektor mit den Aufnahmezeitpunkten aller SAR-Szenen
$x_{(i,j)}$	Pixel mit der Bildkoordinate (i, j)
$\phi_{x_{(i,j)}}^{t_\alpha}$	absolute Phase einer SAR-Szene zum Zeitpunkt t_α an Pixel $x_{(i,j)}$
$\phi_{x_{(i,j)}}^{\Delta t_{\alpha,\beta}}$	absolute Phase eines Interferogramms aus den SAR-Szenen zu den Zeitpunkten t_α und t_β an Pixel $x_{(i,j)}$
$\phi_{x_{(i,j)}}^{\Delta t_{\alpha,\beta} topo}$	Phasenanteil der Topographie der interferometrischen Phase $\phi_{x_{(i,j)}}^{\Delta t_{\alpha,\beta}}$
$\phi_{x_{(i,j)}}^{\Delta t_{\alpha,\beta} flatearth}$	Phasenanteil der flachen Erde der interferometrischen Phase $\phi_{x_{(i,j)}}^{\Delta t_{\alpha,\beta}}$
$\phi_{x_{(i,j)}}^{\Delta t_{\alpha,\beta} defo}$	Phasenanteil der Deformation der (differenziellen) interferometrischen Phase $\phi_{x_{(i,j)}}^{\Delta t_{\alpha,\beta}}$

$\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}} atmo$	Phasenanteil der Atmosphäre der (differentiellen) interferometrischen Phase $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$
$\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}} noise$	Phasenrauschen der (differentiellen) interferometrischen Phase $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$
$\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}} \Delta topo$	topographischer Fehler der differentiellen interferometrischen Phase $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$
$\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}} orbit$	Orbitfehler der differentiellen interferometrischen Phase $\phi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$

Kohärenz

s_{α}	SAR-Signal zum Zeitpunkt t_{α}
γ_c	komplexer Korrelationskoeffizient
γ	Kohärenz
$\hat{\gamma}$	geschätzte Kohärenz
$\gamma_{räumlich}$	räumliche Dekorrelation
$\gamma_{zeitlich}$	zeitliche Dekorrelation
$\gamma_{Doppler}$	Dekorrelation durch die Differenz der Doppler-Frequenzen
γ_{andere}	Zusammenfassung anderer Dekorrelationseffekte

Interferogrammerzeugung

$b_{\perp \alpha}$	orthogonale Basislinie der α -ten SAR-Szene bzgl. einer Referenzszene
$\Delta f_{c \alpha}$	Doppler-Frequenz-Verschiebung zwischen der α -ten SAR-Szene und der Referenzszene
Δt_{α}	zeitliche Basislinie der α -ten SAR-Szene bzgl. einer Referenzszene
P_{α}	Punkt in der $T \times B_{\perp}$ -Ebene, der mit der α -ten SAR-Szene korrespondiert
$\Delta b_{\perp \alpha,\beta}$	relative, orthogonale Basislinie zwischen der α -ten und β -ten SAR-Szene
$\Delta \mathbf{b}_{\perp}$	Vektor der relativen, orthogonalen Basislinien der Interferogramme in einem Bildstapel
$\Delta f_{c \alpha,\beta}$	Doppler-Frequenz-Verschiebung zwischen der α -ten und β -ten SAR-Szene
$\Delta t_{\alpha,\beta}$	relative, zeitliche Basislinie zwischen der α -ten und β -ten SAR-Szene
$\Delta \mathbf{t}$	Vektor der relativen, zeitlichen Basislinien der Interferogramme in einem Bildstapel
$\Delta b_{\perp max}$	Schwellwert für die relative, orthogonale Basislinie bei der Selektion von Datenpaaren
$\Delta f_{c max}$	Schwellwert für die Doppler-Frequenz-Verschiebung bei der Selektion von Datenpaaren
Δt_{max}	Schwellwert für die relative, zeitliche Basislinie bei der Selektion von Datenpaaren
γ^m	<i>stack coherence</i> für den D-InSAR-Bildstapel mit der m-ten SAR-Szene als Referenz
$b_{\perp krit}$	kritische, orthogonale Basislinie
$\Delta f_{c krit}$	kritischer Wert für die Doppler-Frequenz-Verschiebung
Δt_{krit}	kritische, zeitliche Basislinie

Phase Unwrapping als MCF-Problem

$\psi_{x(i,j)}^{\Delta t_{\alpha,\beta}}$	mehrdeutige Phase eines Interferogramms aus den SAR-Szenen zu Zeitpunkt t_{α} und t_{β} an Pixel $x(i,j)$
$k_{x(i,j)}^{\Delta t_{\alpha,\beta}}$	Mehrdeutigkeitsfaktor einer interferometrischen Phase
$\phi_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$	absoluter Phasengradient zwischen zwei Pixeln $x(i,j)$ und $x(i',j')$
$\psi_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$	mehrdeutiger Phasengradient zwischen zwei Pixeln $x(i,j)$ und $x(i',j')$
$k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$	Mehrdeutigkeitsfaktor eines Phasengradienten
$k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta},+}$	positiver Anteil des Mehrdeutigkeitsfaktors eines Phasengradienten
$k_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta},-}$	negativer Anteil des Mehrdeutigkeitsfaktors eines Phasengradienten
$w_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$	Gewicht für eine Kante zwischen zwei Pixeln $x(i,j)$ und $x(i',j')$

EMCF-Algorithmus

θ	Kohärenzschwellwert für die Auswahl der Pixel beim EMCF-Algorithmus
p	prozentualer Anteil für die Auswahl der Pixel beim EMCF-Algorithmus
$\phi_{\Delta x(i,j)(i',j')}^{\Delta t}$	Vektor aller absoluten Phasengradienten zwischen den Pixeln $x(i,j)$ und $x(i',j')$
$\psi_{\Delta x(i,j)(i',j')}^{\Delta t}$	Vektor aller mehrdeutigen Phasengradienten zwischen den Pixeln $x(i,j)$ und $x(i',j')$
$m(\cdot)$	Modell zur Schätzung eines Vektors von absoluten Phasengradienten
$\Delta z_{\Delta x(i,j)(i',j')}$	topographischer Fehler entlang der Kante zwischen den Pixeln $x(i,j)$ und $x(i',j')$
$\Delta v_{\Delta x(i,j)(i',j')}$	Änderung der Deformationsgeschwindigkeit entlang der Kante zwischen den Pixeln $x(i,j)$ und $x(i',j')$
λ	Wellenlänge des Radars
ϑ	lokaler Einfallswinkel
r	Distanz zwischen Sensor und Ziel
$\chi_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$	abgeleitete Beobachtung
$\chi_{\Delta x(i,j)(i',j')}^{\Delta t}$	Vektor von abgeleiteten Beobachtungen
$C_{\Delta x(i,j)(i',j')}$	Gesamtkosten beim Temporal Unwrapping für eine Kante zwischen den Pixeln $x(i,j)$ und $x(i',j')$
$C_{\Delta x(i,j)(i',j')} _{min}$	minimale Gesamtkosten beim Temporal Unwrapping
$\tilde{\phi}_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$	geschätzter, absoluter Phasengradient aus dem Temporal Unwrapping
$\tilde{\phi}_{\Delta x(i,j)(i',j')}^{\Delta t}$	Vektor der geschätzten, absoluten Phasengradienten aus dem Temporal Unwrapping
$\tilde{\chi}_{\Delta x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$	geschätzte, abgeleitete Beobachtung aus dem Temporal Unwrapping
$\tilde{\chi}_{\Delta x(i,j)(i',j')}^{\Delta t}$	Vektor der geschätzten, abgeleiteten Beobachtungen aus dem Temporal Unwrapping
$\tilde{k}_{x(i,j)(i',j')}^{\Delta t_{\alpha,\beta}}$	optimaler Mehrdeutigkeitsfaktor aus dem Temporal Unwrapping
$\tilde{k}_{x(i,j)(i',j')}^{\Delta t}$	Vektor der optimalen Mehrdeutigkeitsfaktoren aus dem Temporal Unwrapping
2^S	maximale Kosten für eine Kante beim Spatial Unwrapping
ρ	Schwellwert der Kostenfunktion beim Spatial Unwrapping

Edgelist-Algorithmus

$k_{\Delta x(i,j)}^{\Delta t(\alpha,\beta)(\gamma,\delta)}$	Mehrdeutigkeitsfaktor einer Kante, die zwischen lagegleichen Pixeln in benachbarten Interferogrammen aufgespannt ist
---	--

SBAS-Methode

i_{master}	Indexvektor für die Masterszenen
i_{slave}	Indexvektor für die Slaveszenen
$d(t_\alpha, i, j)$	Deformation an der Stelle $x(i,j)$ zum Zeitpunkt t_α
$v_{x(i,j)}^{\Delta t_{\alpha,\beta}}$	Phasengeschwindigkeit des Pixels $x(i,j)$ zwischen den benachbarten Aufnahmezeitpunkten t_α und t_β
$v_{x(i,j)}^{\Delta t}$	Vektor von Phasengeschwindigkeiten
B	Designmatrix
B^-	Pseudoinverse der Designmatrix
L	Anzahl der Subsets
$\tilde{v}_{x(i,j)}^{\Delta t_{\alpha,\beta}}$	geschätzte Phasengeschwindigkeit des Pixels $x(i,j)$
$\tilde{v}_{x(i,j)}^{\Delta t}$	Vektor von geschätzten Phasengeschwindigkeiten

B. Abkürzungen

DHM	Digitales Höhenmodell
DIGITAL	Institut für Informations- und Kommunikationstechnologie der Forschungsgesellschaft mbH Joanneum Research in Graz, Österreich
D-InSAR	Differentielle SAR Interferometrie
EMCF	Extended Minimum Cost Flow (3D-Phase Unwrapping Algorithmus)
Envisat	Environmental Satellite (Umweltsatellit der ESA)
ERS-1/ERS-2	European Remote sensing Satellite 1 und 2 (Fernerkundungssatellit der ESA mit SAR-Sensor)
ESA	European Space Agency (Europäische Weltraumbehörde)
InSAR	SAR Interferometrie
MCF	Minimum Cost Flow
NASA	National Aeronautics and Space Administration (US-amerikanische Luft- und Raumfahrtbehörde)
NEST	Next ESA SAR Toolbox (Programm der ESA für die Auswertung von SAR-Daten)
pel	Pixel
PS	Persistent Scatterer oder Permanent Scatterer (langzeitstabile Rückstreuer)
RAR	Real Aperture Radar (Radar mit realer Apertur)
RSB	Remote Sensing software Package Graz (Softwarepaket der Forschungsgesellschaft mbH Joanneum Research in Graz, Österreich)
SAR	Synthetic Aperture Radar (Radar mit synthetischer Apertur)
SLC	Single Look Complex (Dateiformat von SAR-Aufnahmen mit voller Auflösung)
SBAS	Small BAseline Subset (Methode zur Bestimmung von Deformationen aus einem D-InSAR-Bildstapel)
SRTM	Shuttle Radar Topography Mission (Radarmission der NASA zur Oberflächenkartierung)
StaMPS	Stanford Method for Persistent Scatterers (Methode zur Bestimmung von Deformationen bei Verwendung von PS-Punkten)
STUN	Spatio-Temporal Unwrapping Network (Methode zur Bestimmung von Deformationen bei Verwendung von PS-Punkten)
UML	Unified Modeling Language (Modellierungssprache für die Softwareentwicklung)

C. Funktionshandbuch

Das folgende Funktionshandbuch ist eine Dokumentation der Implementierung der hier entwickelten Datenstruktur (vgl. Kapitel 4). Des Weiteren soll es ihre Lauffähigkeit anhand von Realdaten nachweisen. Dazu wurde ein Testdatensatz erzeugt, der in Abschnitt 5.1 vorgestellt wird. Eine Funktionsübersicht mit Kurzbeschreibungen und Referenzen auf diesen Handbuch ist in Abschnitt 5.2 zu finden.

Die Funktionsbeschreibungen in diesem Handbuch beginnen jeweils mit einer Liste, die die Syntax für verschiedene Aufrufmöglichkeiten der Funktion definiert. Anschließend folgt eine genaue Erläuterung darüber, was die verschiedenen Aufrufe bewirken. Diese werden bei einigen Funktionen durch Anmerkungen ergänzt, die auf bestehende Restriktionen der aktuellen Programmversion hinweisen. Gefolgt davon, werden die Eingabe- und Ausgabeparameter dargestellt. Abgeschlossen wird die Beschreibung jeweils mit einem Beispiel, das mit dem Testdatensatz erzeugt wurde. Hier wird jeweils die korrekte Nutzung der Funktion mit einem Auszug aus dem Testskript festgehalten. Im Anschluss daran wird dieser Auszug ausführlich diskutiert.

C.1 Bildstapel()

Syntax:

```
BS = Bildstapel(fp_dinsar, fp_koh, fp_info, date_ref)
```

```
BS = Bildstapel(fp_dinsar, fp_koh, fp_info, date_ref, order)
```

```
BS = Bildstapel(fp_dinsar, fp_koh, fp_info, date_ref, order, theta, p, flag)
```

Beschreibung:

Diese Methode ist der Konstruktor der Klasse *Bildstapel*, mit dem die in dieser Arbeit entwickelte Datenstruktur aufgebaut werden kann. Durch die Überladung ist es möglich, die Struktur in verschiedenen Stufen aufzubauen.

- `BS = Bildstapel(fp_dinsar, fp_koh, fp_info, date_ref)` entspricht dem Aufruf
`BS = Bildstapel(fp_dinsar, fp_koh, fp_info, date_ref, 1)`
- `BS = Bildstapel(fp_dinsar, fp_koh, fp_info, date_ref, order)` baut lediglich die Zeitliche Struktur auf. Dafür wird die Textdatei mit den Basislinieninformationen, wie sie in Abschnitt 2.5.2 beschrieben wurde, sowie das Aufnahmedatum der Referenzszene benötigt.
Für die chronologische Sortierung, die mit dem Parameter `order` möglich ist, wird vorausgesetzt, dass die Dateinamen der Bilder eine bestimmte Form erfüllen. Dies ist nötig, damit die Zeitlichen Kanten den richtigen Dateien zugeordnet werden können. Zulässig sind alle Dateinamen, die bei alphanumerischer Sortierung die Interferogramme und Kohärenzbilder erst nach dem Aufnahmedatum der Slaveszene und anschließend nach dem Datum der Masterszene sortieren. Eine mögliche Form der Bezeichnung wäre `ERS_1997-01-07_36_ERS_1996-09-24_34_sbas_coh.tif`. Dieser Name entspricht dem ersten Kohärenzbild des Testdatensatzes (vgl. Tabelle 5.1).
- Mit `BS = Bildstapel(fp_dinsar, fp_koh, fp_info, date_ref, order, theta, p, flag)` kann entweder zusätzlich die Knotenliste aufgebaut werden oder direkt die gesamte Räumliche Struktur. Dies ist über den Parameter `flag` steuerbar. Die Knotenauswahl ist über den Kohärenzschwellwert `theta` und den prozentualen Anteil `p` festzulegen.

Achtung: In der aktuellen Programmversion werden mit dem Konstruktor `BS = Bildstapel(fp_dinsar, fp_koh, fp_info, date_ref, order, theta, p, flag)` (mit `flag=1`) lediglich die Kanten der Räumlichen Struktur in den Interferogrammen erstellt. In späteren Versionen sollen zusätzlich die Kanten zwischen lagegleichen Knoten von benachbarten Interferogrammen (vgl. Abbildung 3.3) konstruiert werden. Zusätzlich soll von der Funktion die Flächenliste der Räumlichen Struktur aufgebaut werden.

Eingabeparameter:

<code>fp_dinsar</code>	...	string	Dateipfad zum Ordner, in dem die D-InSAR-Bilder liegen.
<code>fp_koh</code>	...	string	Dateipfad zum Ordner, in dem die Kohärenzilder liegen.
<code>fp_info</code>	...	string	Dateipfad und -name der Textdatei, die den Bildstapel beschreibt (vgl. Abschnitt 2.5.2 bzw. 5.1).
<code>date_ref</code>	...	[3x1] double	Vektor mit dem Aufnahmedatum der Referenzszene im Format [JJJJ MM TT].
<code>order</code>	...	[1x1] double	Parameter zur Festlegung der chronologischen Sortierung des Bildstapels. <code>order=1</code> bedeutet, dass die Interferogramme im Bildstapel erst nach dem Aufnahmedatum der Slaveszenen und dann der Masterszenen sortiert werden. <code>order=2</code> sortiert erst nach den Master- und anschließend nach den Slavedaten.
<code>theta</code>	...	[1x1] double	Kohärenzschwellwert für die Knotenauswahl.
<code>p</code>	...	[1x1] double	Prozentualer Anteil für die Knotenauswahl.
<code>flag</code>	...	[1x1] double	Identifizier, der festlegt, ob nur die Knoten eingelesen werden sollen (<code>flag=0</code>) oder ob direkt die ganze Räumliche Struktur aufgebaut werden soll (<code>flag=1</code>).

Ausgabeparameter:

`BS` ... Bildstapel Referenz auf die Instanz der Klasse *Bildstapel*.

Beispiel:

Im Folgenden soll anhand eines Auszuges aus dem Testskript, mit dem die Lauffähigkeit geprüft wurde, veranschaulicht werden, wie der Konstruktor zu verwenden ist.

Listing C.1: Definition der Übergabeparameter und Nutzung verschiedener Konstruktoren zum Aufbau eines Bildstapels.

```

1 %% Settings
2 % Pfad zum Datenordner
3 data_path = 'Testdatensatz\';
4
5 % Pfad zu den Kohärenzbildern
6 data_koh = [data_path, 'Kohaerenz\'];
7 % Pfad zu den Interferogrammen
8 data_dinsar = [data_path, 'DInSAR\'];
9 % Textdatei mit den Informationen zu den Basislinien
10 baseline_info = [data_path, 'Testdatensatz.txt'];
11
12 % Datum der Reserenzsszene [JJJJ MM TT]
13 referenz = [1997 3 18];
14
15 % Sortierung aufsteigend nach dem Datum der Slaveszenen und dann dem Datum der Masterszenen
16 order = 1;

```

```

18 % Kohärenzschwellwert
19 theta = 0.35;
20 % prozentualer Anteil der Pixel, die den Schwellwert in einer Bildsaeule ueberschreiten muessen
21 p = 0.3;

23 % Knoten der Raeumlichen Struktur vermaschen (nein = 0 oder ja = 1)
24 vermaschen = 0;

26 %% Testkript
27 % Aufbau der zeitlichen Struktur
28 BS = Bildstapel(data_dinsar, data_koh, baseline_info, referenz);

30 % Aufbau der Zeitlichen Struktur und einlesen der Knoten der Raeumlichen Struktur
31 BS2 = Bildstapel(data_dinsar, data_koh, baseline_info, referenz, order, theta, p, vermaschen);

```

In den Zeilen 1–24 wurden die Übergabeparameter für den Konstruktor definiert.

In Zeile 28 wird letztlich der Konstruktor aufgerufen, der in diesem Fall lediglich die Zeitliche Struktur aufbaut. Ein alternativer Aufruf des Konstruktors ist in Zeile 31 dargestellt. Hier werden zusätzlich die Knoten der Räumlichen Struktur im Bildstapel eingelesen. Dabei muss der Kohärenzschwellwert von 0,35 in mindestens 30% der Bilder überschritten werden, damit die Pixel in einer Bildsäule als Knoten in die Struktur eingehen.

C.2 leseKnotenEin()

Syntax:

`BS leseKnotenEin(theta, p)`

Beschreibung:

- `BS leseKnotenEin(theta, p)` liest die Knoten der Räumlichen Struktur für den aktuellen Bildstapel `BS` ein. Die Knotenauswahl wird, wie in Abschnitt 3.3 beschrieben, anhand eines Kohärenzschwellwerts `theta` durchgeführt, der in einer Bildsäule von mindestens `p`-100% der Pixel überschritten werden muss.

Anmerkung: Es wird empfohlen, diese Funktion zum Erzeugen der Knotenliste der Räumlichen Struktur zu verwenden, wenn geplant wird, Interferogramme in den Bildstapel einzufügen bzw. aus ihm zu entfernen.

Eingabeparameter:

<code>BS</code>	...	Bildstapel	Referenz auf eine Instanz der Klasse <i>Bildstapel</i> .
<code>theta</code>	...	[1x1] double	Kohärenzschwellwert für die Knotenauswahl.
<code>p</code>	...	[1x1] double	Prozentualer Anteil für die Knotenauswahl.

Beispiel:

Das folgende Listing soll den Unterschied verdeutlichen, der zwischen der Erzeugung der Räumlichen Knotenliste über die Funktion `BS leseKnotenEin(theta, p)` bzw. über den Konstruktor besteht. Die Definition der Übergabeparameter ist dem Listing C.1 zu entnehmen.

Listing C.2: Verschiedene Möglichkeiten zum Einlesen der Knoten der Räumlichen Struktur.

```

1  % Aufbau der zeitlichen Struktur
2  BS = Bildstapel(data_dinsar, data_koh, baseline_info, referenz);

4  % Loesche drei Kanten aus der Zeitlichen Struktur (entferne Bilder aus dem Bildstapel)
5  BS.loescheKante(4,'zeit');
6  BS.loescheKante(8,'zeit');
7  BS.loescheKante(9,'zeit');

9  % Knoten der Raeumlichen Struktur fuer den bearbeiteten Bildstapel einlesen
10 BS.leseKnotenEin(theta, p);

12 % Aufbau der Zeitlichen Struktur und einlesen der Knoten der Raeumlichen Struktur
13 BS2 = Bildstapel(data_dinsar, data_koh, baseline_info, referenz, order, theta, p, vermaschen);

```

In Zeile 2 wird die Zeitliche Struktur über den Konstruktor der Klasse *Bildstapel* aufgebaut. Anschließend werden in Zeile 6–8 drei Interferogramme aus dem Bildstapel entfernt (vgl. Funktionsbeschreibung C.8). Schließlich werden die Knoten für den veränderten Bildstapel in Zeile 11 eingelesen. Mit einem alternativen Konstruktor werden die Knoten in Zeile 15 für gleiches `theta` und `p` für den kompletten Testbildstapel eingelesen.

Das erste Vorgehen führt dazu, dass in jedem Bild 26727 Knoten ausgewählt werden, wohingegen bei der zweiten Variante pro Bild 28826 Knoten erstellt werden.

C.3 vermasche()

Syntax:

```
BS.vermasche()
```

Beschreibung:

- `BS.vermasche()` baut die restliche Räumliche Struktur auf, sofern bereits die Knoten für kohärente Pixel im D-InSAR-Bildstapel erzeugt wurden.

Achtung: In der aktuellen Programmversion werden mit der Funktion `BS.vermasche()` lediglich die Kanten der Räumlichen Struktur in den Interferogrammen erstellt. In späteren Versionen sollen zusätzlich die Kanten zwischen lagegleichen Knoten von benachbarten Interferogrammen (vgl. Abbildung 3.3) konstruiert werden. Zusätzlich soll von der Funktion die Flächenliste der Räumlichen Struktur aufgebaut werden.

Eingabeparameter:

BS ... Bildstapel Referenz auf eine Instanz der Klasse *Bildstapel*.

Beispiel:

Mit dem Listing C.3 soll die korrekte Nutzung der hier beschriebenen Funktion demonstriert werden. Die in diesem Auszug aus dem Testskript nicht definierten Variablen sind dem Listing C.1 zu entnehmen.

Listing C.3: Erzeugung der Vermaschung für einen bearbeiteten Bildstapel

```

1 % Aufbau der Zeitlichen Struktur
2 BS = Bildstapel(data_dinsar, data_koh, baseline_info, referenz);

4 % Loesche drei Kanten aus der Zeitlichen Struktur (entferne Bilder aus dem Bildstapel)
5 BS.loescheKante(4,'zeit');
6 BS.loescheKante(8,'zeit');
7 BS.loescheKante(9,'zeit');

9 % Knoten der Raeumlichen Struktur fuer den bearbeiteten Bildstapel einlesen
10 BS.leseKnotenEin(theta, p);

12 % Erzeugung der Kanten in der Raeumlichen Struktur
13 BS.vermasche();

```

Zuerst wird in den Zeilen 1–8 die Zeitliche Struktur aufgebaut, aus der anschließend drei Kanten entfernt werden. Für den so manipulierten Bildstapel werden die Knoten der räumlichen Struktur eingelesen (Zeile 11). Schließlich werden diese in Zeile 14 vermascht.

C.4 scatter()

Syntax:

```
h = BS.scatter(ID)
```

```
h = BS.scatter(ID, theta, p)
```

Beschreibung:

Diese Funktion erstellt einen Scatter-Plot, in dem die Knoten der Räumlichen Struktur für ein Interferogramm dargestellt werden. Knoten, die den Kohärenzschwellwert θ unterschreiten, werden rot eingefärbt. Die anderen Knoten werden blau markiert.

Zusätzlich können auch restriktivere Parameter θ und p für Knotenauswahl übergeben werden als bei der Erzeugung der Räumlichen Knotenliste. In diesem Fall werden nur diejenigen Knoten geplottet, die diese Kriterien erfüllen.

- `h = BS.scatter(ID)` stellt die aktuelle Knotenauswahl für das Interferogramm dar, das mit der Kante ID der Zeitlichen Struktur assoziiert werden kann.
- Mit `h = BS.scatter(ID, theta, p)` wird die Knotenauswahl für die Parameter `theta` und `p` dargestellt. Dabei müssen die beiden Parameter größer oder gleich den Werten sein, die beim Einlesen der Knoten verwendet wurden.

Eingabeparameter:

BS	...	Bildstapel	Referenz auf eine Instanz der Klasse <i>Bildstapel</i> .
ID	...	[1x1] double	Kanten-ID der Zeitlichen Struktur.
theta	...	[1x1] double	Kohärenzschwellwert für die Knotenauswahl.
p	...	[1x1] double	Prozentualer Anteil für die Knotenauswahl.

Ausgabeparameter:

h ... [1x1] double Referenz auf das erzeugte *figure*-Objekt.

Beispiel:

Im Listing C.4 sollen die beiden Varianten vorgestellt werden, die oben für die Funktion `h = BS.scatter(ID)` beschrieben wurden. Die nicht definierten Übergabeparameter sind bei Interesse dem Listing C.1 zu entnehmen.

Listing C.4: Nutzungsmöglichkeiten der Scatter-Funktion der Klasse *Bildstapel*.

```

1 % Aufbau der Zeitlichen Struktur
2 BS = Bildstapel(data_dinsar, data_koh, baseline_info, referenz);

4 % Kohärenzschwellwert
5 theta = 0.35;
6 % prozentualer Anteil der Pixel, die den Schwellwert in einer Bildsaeule ueberschreiten muessen
7 p = 0.3;

9 % Knoten der Raeumlichen Struktur einlesen
10 BS.leseKnotenEin(theta, p);

12 % Plot des Basisliniennetzwerkes
13 BS.plot('zeit');

15 % Plotten der Knotenverteilung in der Azimuth-Range-Ebene fuer das Bild, das mit der Kante 6 in
16 % der Zeitlichen Struktur korrespondiert.
17 BS.scatter(6);

19 % Plotten der Knotenauswahl fuer dasselbe Bild mit anderem Kohärenzschwellwert theta_neu und
20 % anderem prozentualen Anteil p_neu
21 theta_neu = 0.4;
22 p_neu = 0.35;
23 BS.scatter(6, 0.4, 0.35);

```

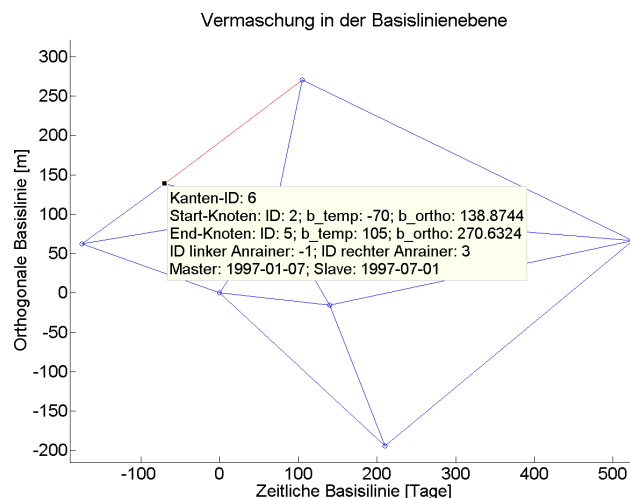


Abbildung C.1: Nachweis des Zusammenhangs zwischen Kante 6 des Basisliniennetzwerkes und dem Interferogramm mit der Masterszene vom 07.01.1997 und der Slaveszene vom 01.07.1997.

In den Zeilen 1–10 wird zunächst das Basisliniennetzwerk aufgebaut und die Knoten der Räumlichen Struktur eingelesen, wobei der Kohärenzschwellwert mit $\theta = 0,35$ und der prozentuale Anteil der Pixel, die diesen

Wert überschreiten müssen, auf 30% festgelegt werden. Anschließend wird in Zeile 13 der Graph in der $T \times B_{\perp}$ -Ebene geplottet (vgl. Funktionsbeschreibung C.5).

In diesem Plot wurden die Informationen zur Kante mit der $ID = 6$ abgerufen. Das Ergebnis dieser Abfrage ist in Abbildung C.1 dargestellt, in der die Kante rot markiert ist. Aus dem Informationskasten lässt sich ablesen, dass diese Kante mit dem Interferogramm korrespondiert, das aus den ERS-Aufnahmen vom 07.01.1997 (Master) und vom 01.07.1997 (Slave) erzeugt wurde.

Die aktuelle Knotenauswahl für dieses Interferogramm wird in Zeile 17 mit der Scatter-Methode dargestellt (vgl. Abbildung C.2a). Anschließend werden in Zeile 21 und 22 ein alternativer Kohärenzschwellwert mit $\theta = 0,4$ festgelegt und der zu erfüllende prozentuale Anteil auf 35% erhöht. Der Einfluss dieser Parameter auf die Knotenauswahl wird in Zeile 23 visualisiert (vgl. Abbildung C.2b).

Wie den Abbildungen C.1 und C.2 zu entnehmen ist, lässt sich die hier beschriebene Funktion für verschiedene Zwecke nutzen. Zum einen ist es durch die Einfärbung der Knoten möglich, die Qualität eines Interferogramms optisch abzuschätzen. Ein hoher Anteil von roten Punkten deutet darauf hin, dass das betrachtete Interferogramm im Vergleich zu den anderen D-InSAR-Bildern im Stapel ein höheres Phasenrauschen aufweist (vgl. Abschnitt 2.4). Zum anderen kann untersucht werden, wie die Parameter θ und p festzulegen sind, damit nicht zu viele Informationen verloren gehen.

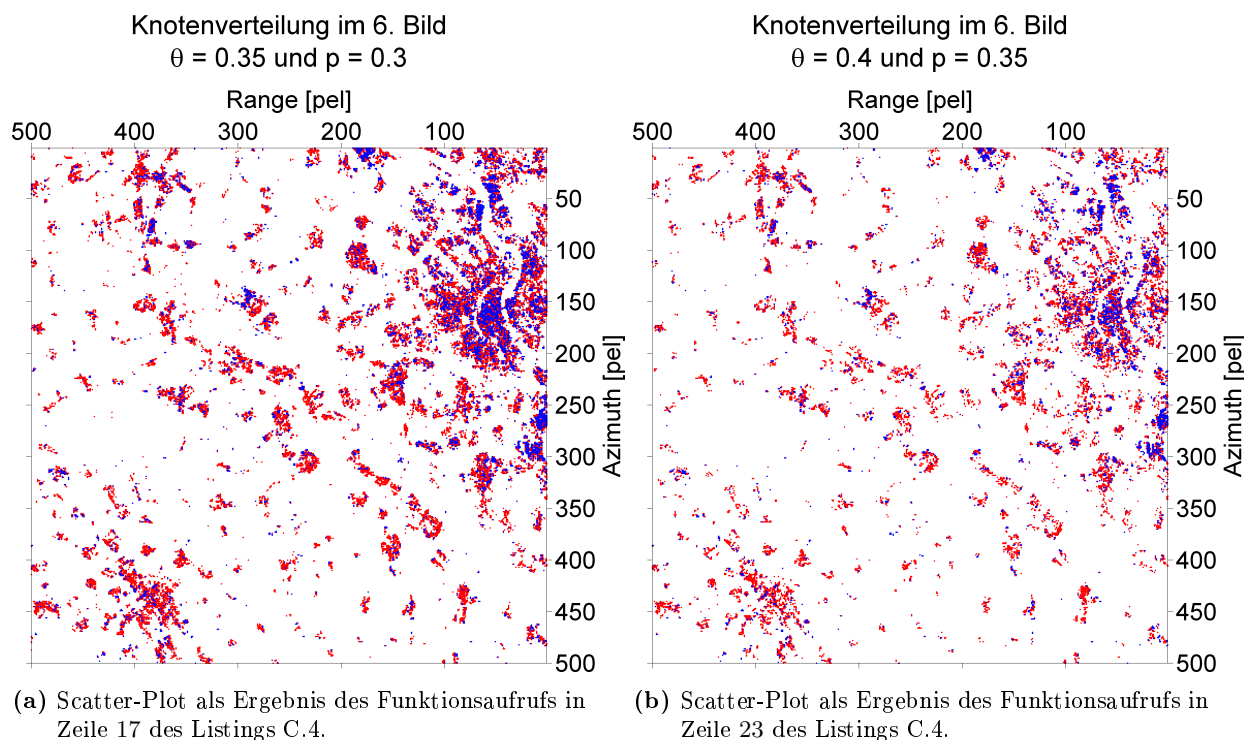


Abbildung C.2: Beispiele für die verschiedenen Nutzungsmöglichkeiten der Scatter-Funktion der Klasse *Bildstapel*.

C.5 plot()


Syntax:

```
h = BS.plot(struktur)
```

```
h = BS.plot(struktur, ID)
```

Beschreibung:

Diese Funktion erzeugt einen interaktiven Plot des Basisliniennetzwerkes beziehungsweise der Vermaschung

der Räumlichen Struktur in einem ausgewählten Interferogramm. Mit dem sogenannten *DataCursor* ¹ können durch Anklicken der Objekte Informationen zu diesen abgerufen werden.

Für den Graphen in der $T \times B_{\perp}$ -Ebene können Informationen zu den Knoten und den Kanten angezeigt werden. Für die Plots in der $A_Z \times R_G$ -Ebene sind nur Angaben zu den Kanten abfragbar. Da für die Zuordnung der Daten zu den Graphenelementen jedes Objekt separat geplottet werden muss, ist es auf Grund der Menge der Objekte nicht möglich, auch die Knoten in der $A_Z \times R_G$ -Ebene darzustellen.

- `h = BS.plot(struktur)` erzeugt einen interaktiven Plot des Basisliniennetzwerkes für `struktur='zeit'`.
- `h = BS.plot(struktur, ID)` erzeugt für `struktur='raum'` einen interaktiven Plot der Räumlichen Struktur in dem Interferogramm, das der Kanten-ID des Basisliniennetzwerkes zugeordnet ist.

Eingabeparameter:

BS	...	Bildstapel	Referenz auf eine Instanz der Klasse <i>Bildstapel</i> .
struktur	...	string	Der String gibt an, ob ein Plot für die Zeitliche Struktur (<code>struktur='zeit'</code>) erstellt werden soll oder für eine Bildebene der Räumlichen Struktur (<code>struktur='raum'</code>).
ID	...	[1x1] double	Kanten-ID der Zeitlichen Struktur, die mit dem Interferogramm korrespondiert, für das der Plot erzeugt werden soll.

Ausgabeparameter:

h	...	[1x1] double	Referenz auf das erzeugte <i>figure</i> -Objekt.
---	-----	--------------	--

Beispiel:

Mit dem Listing C.5 soll die korrekte Nutzung der Plot-Funktion dokumentiert werden. Das Beispiel knüpft an die Festlegungen aus Listing C.1 an.

Listing C.5: Korrekte Nutzung der Plot-Funktion der Klasse *Bildstapel*.

```

1 % Aufbau der Zeitlichen Struktur
2 BS = Bildstapel(data_dinsar, data_koh, baseline_info, referenz);

4 % Plot des Basisliniennetzwerkes
5 BS.plot('zeit');

7 % Loesche Kante aus der Zeitlichen Struktur (entferne Bilder aus dem Bildstapel)
8 BS.loescheKante(4,'zeit');

10 % Knoten der Raeumlichen Struktur fuer den bearbeiteten Bildstapel einlesen
11 BS.leseKnotenEin(theta, p);

13 % Erzeugung der Kanten in der Raeumlichen Struktur
14 BS.vermasche();

16 % Plot der Vermaschung in der Azimuth-Range-Ebene fuer das Bild, das mit
17 % der Kante 6 in der Zeitlichen Struktur korrespondiert
18 BS.plot('raum', 6);

```

¹Icon aus der Nutzeroberfläche von Matlab ©MathWorks Inc.

Das oben aufgeführte Beispiel beginnt mit der Erzeugung der zeitlichen Struktur (Zeile 2), die in Zeile 5 mit der hier vorgestellten Plot-Funktion visualisiert wird. In der erstellten Abbildung C.3 werden beispielhaft die Informationen zu einem Knoten und einer Kante abgefragt.

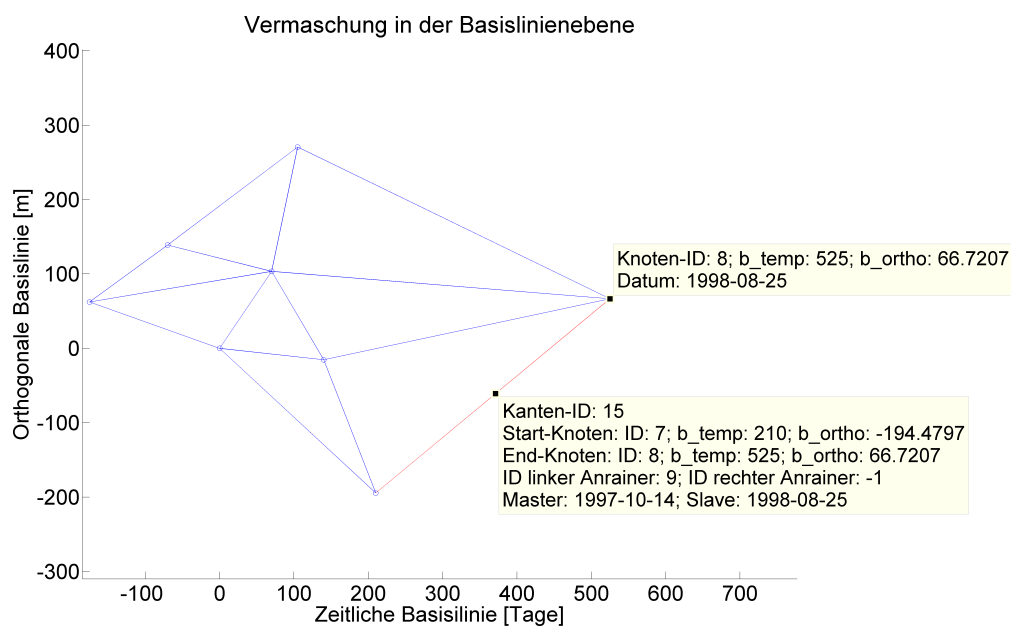


Abbildung C.3: Informationsabfrage im Plot des Basisliniennetzwerkes.

Die Informationen, die zu einem Element angezeigt werden, können der entnommen werden. Die Liste der Daten zu einer Kante wurde bewusst ausführlich gestaltet, sodass alle mit ihr verknüpften Elemente leicht identifiziert werden können. Dazu gehören nicht nur die begrenzenden Knoten, sondern auch die angrenzenden Flächen (Anrainer) und das zugeordnete Interferogramm (über die Aufnahmedaten der Master- und Slaveszene). Es sei noch angemerkt, dass die $ID = -1$ beim rechten Anrainer für die Masche *Außen* steht. Zur besseren Identifikation beim Anklicken von Kantenobjekten, werden diese rot eingefärbt. Knoten werden automatisch durch den Datencursor hervorgehoben.

Nach dem Plot des Basisliniennetzwerkes wird aus diesem eine Kante gelöscht (Zeile 8). Anschließend werden die Knoten der Räumlichen Struktur eingelesen und vermascht (Zeile 11 und 14). Schließlich wird diese Vermaschung in Zeile 18 für das D-InSAR-Bild dargestellt, das mit der Kanten-ID 6 des Zeitlichen Graphen korrespondiert. In dem erzeugten Plot wird exemplarisch die Information zu einer Kante abgefragt (vgl. Abbildung C.4).

Auch in den so erstellten Plots wird die angeklickte Kante für eine leichtere Zuordnung rot eingefärbt.

Zurzeit umfassen die angezeigten Informationen lediglich die Kanten-ID sowie die IDs und Koordinaten der Anfangs- und Endknoten. In diesem Fall sind die Koordinaten von besonderem Interesse, da sie die einzige Möglichkeit darstellen, um Knoten in der Räumlichen Struktur zu identifizieren. In späteren Programmversionen soll die Liste noch um die Angabe der Anrainer-IDs erweitert werden.

Am Titel des Bildes ist zu erkennen, dass die Kanten-ID im Basisliniennetzwerk nicht zwangsläufig mit der Position im Bildstapel übereinstimmt. Das Interferogramm, das der $ID = 6$ zugeordnet werden kann, steht an fünfter Stelle im Bildstapel. Dies ist auf das Löschen der Kante mit der $ID = 4$ in Zeile 8 des Listings C.5 zurückzuführen.

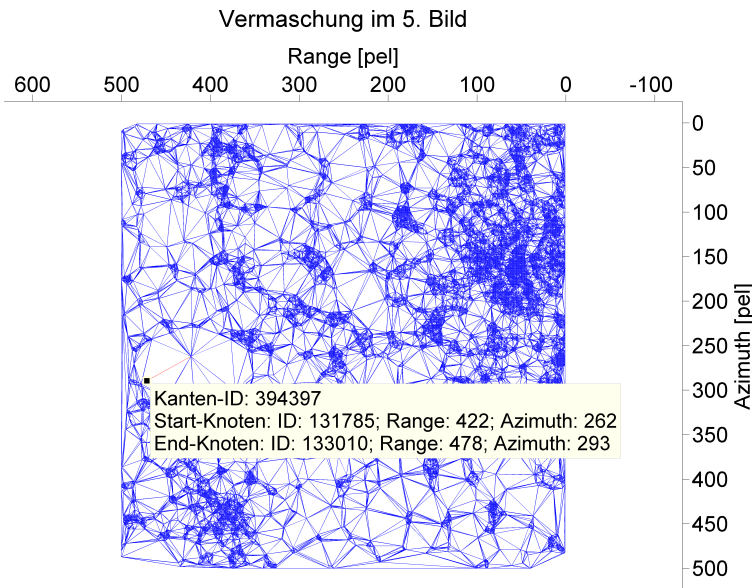


Abbildung C.4: Informationsabfrage in der Vermaschung der Knoten in einem Interferogramm.

C.6 showInfo()

Syntax:

`BS.showInfo(ID, typ, struktur)`

Beschreibung:

Mit dieser Funktion können die Informationen zu einem Knoten-, Kanten- oder Flächenobjekt aus der Zeitlichen bzw. Räumlichen Struktur in der Konsole ausgegeben werden.

- `BS.showInfo(ID, typ, struktur)` erzeugt für das Objekt mit der ID von der Art `typ` aus der Struktur `struktur` eine geordnete Konsolenausgabe.

Achtung: In der aktuellen Programmversion existiert noch keine Flächenliste der Räumlichen Struktur. Aus diesem Grund ist ein Funktionsaufruf der Art `BS.showInfo(ID, 'f', 'raum')` nicht möglich.

Eingabeparameter:

<code>BS</code>	...	<code>Bildstapel</code>	Referenz auf eine Instanz der Klasse <i>Bildstapel</i> .
<code>ID</code>	...	<code>[1x1] double</code>	ID des Elements, dessen Informationen in der Konsole ausgegeben werden soll.
<code>typ</code>	...	<code>char</code>	Der Character gibt an, ob es sich bei dem Element um einen Knoten (<code>typ='v'</code>), eine Kante (<code>typ='e'</code>) oder eine Fläche (<code>typ='f'</code>) handelt.
<code>struktur</code>	...	<code>string</code>	Der String gibt an, ob das Element aus der Zeitlichen Struktur (<code>struktur='zeit'</code>) oder aus der Räumlichen Struktur (<code>struktur='raum'</code>) stammt.

Beispiel:

Als Fortführung des Beispiels aus der Funktionsbeschreibung C.5 werden im Listing C.6 einige Konsolenausgaben für Objekte erzeugt, die auch in den Abbildungen C.3 und C.4 dargestellt sind.

Listing C.6: Erzeugen von Konsolenausgaben.

```

1 % Konsolenausgaben fuer verschiedene Element der Zeitlichen Struktur
2 BS.showInfo(8, 'v', 'zeit');      % Knoten --> 'v' fuer Vertex
3 BS.showInfo(15, 'e', 'zeit');    % Kante  --> 'e' fuer Edge
4 BS.showInfo(9, 'f', 'zeit');    % Knoten --> 'f' fuer Face

6 % Konsolenausgaben fuer verschiedene Element der Raeumlichen Struktur
7 BS.showInfo(131785, 'v', 'raum'); % Knoten
8 BS.showInfo(394397, 'e', 'raum'); % Kante

```

In Zeile 2 werden die Informationen zu dem Knoten des Basisliniennetzwerkes ausgegeben, der auch in Abbildung C.3 mit dem Datencursor ausgewählt wurde. Die Informationen im Plot sind daher auch in der folgenden Konsolenausgabe zu finden:

ID	b_temp	b_ortho	f_doppler	Datum	ausgehende Kanten
8	525	66.721	38.062	1998-08-25	12, 13, 14, 15

Für einen Knoten des Basisliniennetzwerkes werden seine ID, die zeitliche (in Tagen) und orthogonale (in m) Basislinie bezüglich der Referenzszenes sowie die Doppler-Frequenz-Verschiebung (in Hz) ausgegeben. Zusätzlich wird das Aufnahmedatum der SAR-Szene, die dem Knoten zugeordnet werden kann, zurückgegeben. Abgeschlossen wird die Liste mit den IDs der ausgehenden Kanten.

Der Funktionsaufruf in Zeile 3 erzeugt die Konsolenausgabe für die in Abbildung C.3 markierte, rote Kante:

ID	ID_Anfang	ID_Ende	db_temp	db_ortho	df_doppler	linker Anrainer	rechter Anrainer	D-InSAR Bild	Kohaerenzbild
15	7	8	315	261.241	23.547	9			

-1 | D-InSAR Pfad | Kohärenz Pfad

Aus Darstellungsgründen wurde die Ausgabe in zwei Blöcke geteilt. Die unteren zwei Zeilen werden eigentlich rechts neben den oberen zwei Zeilen ausgegeben. Außerdem werden die eigentlichen Dateipfade zu dem Interferogramm und Kohärenzbild durch die Platzhalter `D-InSAR Pfad` und `Kohärenz Pfad` ersetzt.

Neben der ID der Kante werden die IDs ihres Anfangs- und Endknotens für die Richtungsbestimmung ausgegeben. Es folgen die relativen Basislinien und die Doppler-Zentroid-Differenz (Einheiten wie oben). Zusätzlich sind die IDs der Anrainer sowie die Dateipfade und -namen zu den Bildern, die mit der Kante korrespondieren, im Output enthalten.

Als letztes Element der Zeitlichen Struktur wird eine Flächeninformation ausgegeben (Zeile 4). Bei der Fläche handelt es sich um den linken Anrainer der zuvor ausgegebenen Kante.

ID	ID_Kanten
9	11, -14, 15

Der Output für die Flächen ist relativ kurz. Er umfasst die Flächen-ID sowie eine Liste mit den IDs der Kanten, die die Flächen begrenzen. Die Kanten in dieser List bilden stets einen positiv orientierten Zyklus, daher haben einige IDs ein negatives Vorzeichen. Dies bedeutet, dass die Kante entgegen ihrer Richtung durchlaufen werden muss, die durch den Start- und Endknoten festgelegt ist.

In Zeile 7 wird eine Konsolenausgabe für einen Knoten der räumlichen Struktur erzeugt. Bei dem ausgewählten Knoten handelt es sich um den Startknoten der Kante, die in Abbildung C.4 ausgewählt wurde.

Kohaerenzschwellwert $\theta = 0.35$

prozentualer Anteil $p = 30\%$

ID	p_r	p_a	I	psi	gamma	ref_Bild	ref_Lage	ausgehende Kanten
131785	422	262	2.56	-2.61291099	0.09	5	24877	385910, 387088,...

Für die Knoten der Räumlichen Struktur werden als einzige zusätzliche Informationen in Form eines Headers in die Konsole geschrieben. In diesem werden die Parameter der Knotenauswahl θ und p angezeigt. Die Ausgabe für den eigentlichen Knoten umfasst neben seiner ID seine Pixelkoordinaten in Range- und Azimuthrichtung sowie den Intensitätswert, den mehrdeutigen Phasenwert und den Kohärenzwert. Zusätzlich wird eine Referenz auf das Interferogramm im Bildstapel bzw. auf alle Knoten in diesem Bild angegeben. Die nächste Referenz zeigt auf eine Liste mit den IDs aller Knoten, die die selben Pixelkoordinaten haben. Als letztes wird eine Liste mit den IDs der ausgehenden und ankommenden Kanten dargestellt. Aus Platzgründen wurde diese hier nach dem zweiten Element abgebrochen.

Als letztes werden in Zeile 8 die Informationen zu der ausgewählten Kante in Abbildung C.4 in der Konsole ausgegeben.

```
ID      | ID_Anfang | ID_Ende | delta_psi | ref_Bild | ref_Lage | l. Anr. | r. Anr.
394397 | 131785   | 133010 | -2.22956700 | 5 | 74553 | NaN | NaN
```

Für die Kanten der Räumlichen Struktur werden ihre eigene ID sowie die IDs der begrenzenden Knoten ausgegeben. Als abgeleiteten Wert enthält die Ausgabe außerdem den mehrdeutigen Phasengradienten, gefolgt von einigen Referenzen. Die erste stellt eine Beziehung zum Interferogramm her, in dem die Kante liegt, sowie zu allen anderen Kanten, die in diesem D-InSAR-Bild liegen. Darauf folgt eine Referenz auf alle lagegleichen Kanten. Zuletzt sollen eigentlich noch die Anrainer referenziert werden, jedoch existiert in der aktuellen Programmversion bisher keine Räumliche Flächenklasse. Daher werden hier die *NaN*-Werte ausgegeben.

C.7 saveInfo()

Syntax:

```
BS.saveInfo(fp, typ, struktur)
```

Beschreibung:

Mit dieser Funktion können die Informationen zu allen Knoten-, Kanten- oder Flächenobjekt aus der Zeitlichen bzw. Räumlichen Struktur in eine Textdatei geschrieben werden.

- `BS.saveInfo(fp, typ, struktur)` erzeugt die Textdatei `fp` mit den Informationen zu allen Objekten der Art `typ` aus der Struktur `struktur`.

Achtung: In der aktuellen Programmversion existiert noch keine Flächenliste der Räumlichen Struktur. Aus diesem Grund ist ein Funktionsaufruf der Art `BS.saveInfo(fp, 'f', 'raum')` nicht möglich.

Hinweis: Das Schreiben der Textdatei für die Knoten der Räumlichen Struktur nimmt viel Zeit in Anspruch, da die Informationen in einer For-Schleife über alle Listeneinträge für jedes Element separat geschrieben werden müssen. Dies ist auf die variierenden Längen der Listen mit den ausgehenden Kanten zurückzuführen.

Eingabeparameter:

BS	...	Bildstapel	Referenz auf eine Instanz der Klasse <i>Bildstapel</i> .
fp	...	string	Dateipfad und -name der Textdatei, in die die Informationen geschrieben werden sollen.
typ	...	char	Der Character gibt an, ob es sich bei dem Element um einen Knoten (<code>typ='v'</code>), eine Kante (<code>typ='e'</code>) oder eine Fläche (<code>typ='f'</code>) handelt.
struktur	...	string	Der String gibt an, ob das Element aus der Zeitlichen Struktur (<code>struktur='zeit'</code>) oder aus der Räumlichen Struktur (<code>struktur='raum'</code>) stammt.

Beispiel:

Das folgende Listing C.7 ist ein Auszug aus dem Testskript, mit dem die Lauffähigkeit der implementierten Datenstruktur nachgewiesen wird. In ihm wird dokumentiert, wie Textdateien für die einzelnen Listen der Struktur erstellt werden können. Für die Erzeugung der Instanz BS der Klasse Bildstapel sei auf Listing C.1 verwiesen.

Listing C.7: Erzeugung von Textdateien mit Informationen zu den Objekten der Datenstruktur.

```
1 % Speichern der Informationen zu allen Elementen der Zeitlichen Struktur
2 BS.saveInfo('Knoteninfo_zeit.txt', 'v', 'zeit');
3 BS.saveInfo('Kanteninfo_zeit.txt', 'e', 'zeit');
4 BS.saveInfo('Flaecheninfo_zeit.txt', 'f', 'zeit');
6 % Speichern der Informationen zu allen Elementen der Raeumlichen Struktur
7 BS.saveInfo('Knoteninfo_raum.txt', 'v', 'raum');
8 BS.saveInfo('Kanteninfo_raum.txt', 'e', 'raum');
```

Die Beschreibung des Inhalts der Dateien, die in Zeile 2–8 erzeugt werden, kann dem Beispiel aus der Funktionsbeschreibung C.6 entnommen werden. Der Unterschied zu den dortigen Angaben besteht darin, dass die Informationen sequentiell für jedes Element einer Liste in die Datei geschrieben werden.

C.8 *loescheKante()*

Syntax:

`BS.loescheKante(ID, struktur)`

Beschreibung:

Mit dieser Funktion soll es möglich sein, eine Kante ordnungsgemäß aus der Zeitlichen bzw. Räumlichen Struktur zu löschen. Dabei umfasst diese Operation nicht nur die Entfernung des Objektes aus der entsprechenden Liste.

Im Folgenden soll der zukünftige Umfang der Funktion erläutert werden, obwohl momentan nur das Löschen aus der Zeitlichen Struktur implementiert ist.

Grenzen zwei Flächen an die zu entfernende Kante, so werden diese durch die Funktion vereinigt. Wenn sie jedoch nur einen Anrainer hat, also Teil der konvexen Hülle ist, wird die angrenzende Fläche gelöscht. In der Räumlichen Struktur werden dann zusätzlich alle Kanten der Fläche gelöscht, die ebenfalls Teil der konvexen Hülle sind. So soll sichergestellt werden, dass die Vermaschung der Knoten in den Interferogrammen nur geschlossene Zyklen enthält.

Zusätzlich werden alle Verwaltungsstrukturen aktualisiert, sodass keine Referenzen auf nicht existierende Elemente vorkommen.

Das Löschen einer Kante aus der Zeitlichen Struktur ist gleichzusetzen mit dem Entfernen des zugeordneten Interferogramms aus dem Bildstapel. Entsprechend sind alle Objekte (Knoten, Kanten und Flächen) der Räumlichen Struktur, die in diesem Interferogramm liegen, aus der Datenstruktur zu entfernen.

Wird eine Kante aus der Vermaschung in einem der Interferogramme gelöscht, so sind ebenfalls alle lagegleiche Kanten in den anderen D-InSAR-Bildern zu entfernen. Auf diese Weise wird garantiert, dass die Vermaschungen in jedem Bild identisch ist.

Ein Teil der hier beschriebenen Konsequenzen beim Entfernen einer Kante, sind in der aktuelle Programmversion noch nicht umgesetzt. Für eine korrekte Nutzung der Funktion sei hier insbesondere auf die unten aufgeführten Anmerkungen sowie das Beispiel verwiesen.

- `BS.loescheKante(ID, struktur)` löscht die Kante mit der ID aus der angegebenen `struktur`.

Achtung: In der aktuellen Programmversion ist es noch nicht möglich, Kanten aus der Räumlichen Struktur zu löschen. Der Funktionsaufruf der Form `BS.loescheKante(ID, 'raum')` hat daher keinen Effekt auf die Datenstruktur.

Hinweis: Der Funktionsaufruf der Form `BS.loescheKante(ID, 'zeit')` führt in der momentanen Programmversion nicht zur Entfernung der betroffenen Objekte aus der Räumlichen Struktur. Ist eine Nachbearbeitung des Bildstapels gewünscht, sollte daher die Räumliche Struktur erst nach dem Entfernen von Interferogrammen aus dem Bildstapel aufgebaut werden (vgl. Funktionsbeschreibungen C.1, C.2 und C.3). Wenn zur Bestimmung der zu löschenden Kante eine optische Untersuchung der D-InSAR-Aufnahmen mit der Scatter-Funktion (vgl. Funktionsbeschreibung C.4) durchgeführt werden soll, wird folgendes Vorgehen vorgeschlagen. Es sollte ein Bildstapel mit den Knoten der Räumlichen Struktur aufgebaut werden. Anschließend können mit der Scatter-Funktion die Kanten identifiziert werden, die aus dem Basisliniennetzwerk entfernt werden sollen. Nun soll der Bildstapel erneut aufgebaut werden, wobei vorerst nur der zeitliche Graph erzeugt werden sollte. Dort können die ausgesuchten Kanten entfernt und abschließend kann die Räumliche Struktur aufgebaut werden.

Eingabeparameter:

BS	...	Bildstapel	Referenz auf eine Instanz der Klasse <i>Bildstapel</i> .
ID	...	[1x1] double	ID der Kante, die gelöscht werden soll.
struktur	...	string	Der String gibt an, ob die Kante aus der Zeitlichen Struktur (<code>struktur='zeit'</code>) oder aus der Räumlichen Struktur (<code>struktur='raum'</code>) entfernt werden soll.

Beispiel:

Mit dem Beispiel in Listing C.8 wird veranschaulicht, wie sich das Löschen von Kanten auf andere Elemente, also Knoten und Flächen auswirkt. Die Parameter zum Aufbau des Bildstapels sind in Listing C.1 zu finden.

Listing C.8: Auswirkung des Löschens von Kanten auf andere Objekte.

```

1  % Aufbau der Zeitlichen Struktur
2  BS = Bildstapel(data_dinsar, data_koh, baseline_info, referenz);

4  % Plot des Basisliniennetzwerkes vor der Bearbeitung
5  BS.plot('zeit');

7  % Es soll die Kante mit der ID=12 aus der Zeitlichen Struktur geloescht werden.

9  % Ausgeben der Zyklen der Anrainer der Kante 12 (Informationen aus dem Plot)
10 BS.showInfo(5, 'f', 'zeit');
11 BS.showInfo(7, 'f', 'zeit');

13 % Ausgeben der Informationen des Start- und Endknotens der Kante 12
14 BS.showInfo(4, 'e', 'zeit');
15 BS.showInfo(8, 'e', 'zeit');

17 % Loeschen der Kante
18 BS.loescheKante(12, 'zeit');

20 % Ausgeben der Informationen des Start- und Endknotens nach dem Loeschen der Kante 12
21 BS.showInfo(4, 'e', 'zeit');
22 BS.showInfo(8, 'e', 'zeit');
```

```

24 % Plot des Basisliniennetzwerkes nach der Bearbeitung
25 BS.plot('zeit');

27 % Ausgeben des Zyklus der vereinigten Masche
28 BS.showInfo(5, 'f', 'zeit');

30 % Loeschen weiterer Kanten, sodass eine "baumelnde" Kante entsteht
31 BS.loescheKante(10, 'zeit');
32 BS.loescheKante(15, 'zeit');

34 % Plot des Basisliniennetzwerkes nach der Bearbeitung
35 BS.plot('zeit');

37 % Ausgabe der Informationen zur "baumelnden" Kante mit der ID=11
38 BS.showInfo(11, 'e', 'zeit');

```

In Zeile 2 wird mit dem Konstruktor der Klasse *Bildstapel* die Zeitliche Struktur aufgebaut. Anschließend wird diese in Zeile 5 geplottet. Das Ergebnis ist in Abbildung C.5 dargestellt, in der die Kante mit der $ID = 12$ markiert ist. Diese Kante soll als erstes aus dem Graphen entfernt werden. Vorerst werden jedoch

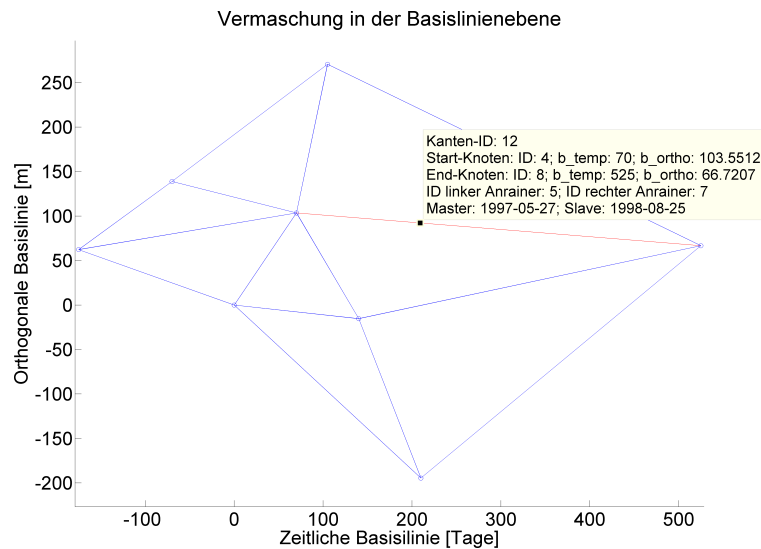


Abbildung C.5: Interaktiver Plot mit Informationsabruf zur Kante mit der $ID = 12$.

die Informationen zu den Anrainern der Kante in Zeile 10–11 ausgegeben, um die Definition der Zyklen für diese Maschen abzufragen. Die Konsolenausgabe in zusammengefasster Form enthält folgende Angaben:

```

ID | ID_Kanten
 5 | -7, 12, -13
 7 | 9, -12, 14

```

Anschließend werden auch die Informationen zum Anfangs- und Endknoten ausgegeben (Zeile 14–15), die hier auf die Knoten-ID und die Liste der ausgehenden Kanten reduziert wurde:

```

ID | ... | ausgehende Kanten
 4 | ... | 3, 4, 5, 7, 9, 12
 8 | ... | 12, 13, 14, 15

```

Nun wird schließlich die Kante in Zeile 18 gelöscht. Der Einfluss auf die Liste der ausgehenden Kanten wird durch die Funktionsaufrufe in Zeile 21 und 22 dokumentiert, die folgende (hier gekürzte) Ausgabe erzeugen:

```
ID | ... | ausgehende Kanten
4  | ... | 3, 4, 5, 7, 9
8  | ... | 13, 14, 15
```

Wie hier zu sehen ist, wird die Korrektur der Listen korrekt durchgeführt.

In Zeile 25 wird nun das veränderte Basisliniennetzwerk dargestellt. In dem Plot werden zwei Kanten markiert. Anhand der Informationsanzeige kann die ordnungsgemäße Vereinigung der Anrainer der gelöschten Kante zu einer Fläche nachgewiesen werden (vgl. Abbildung C.6).

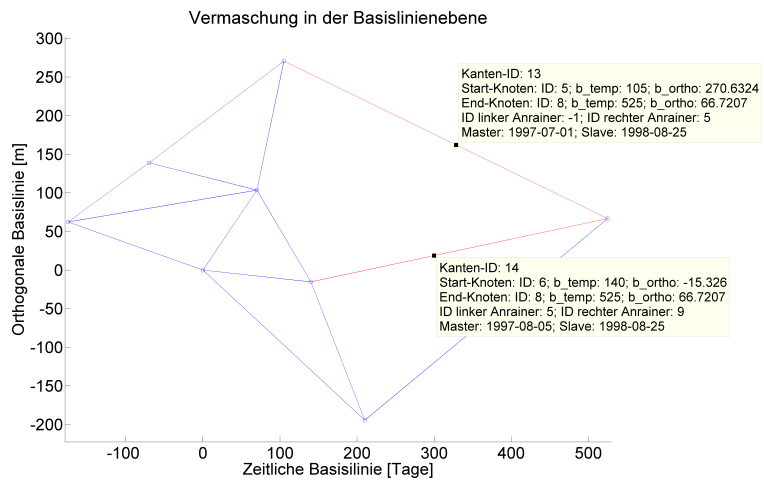


Abbildung C.6: Interaktiver Plot mit Informationsabruf zu den Kante mit der $ID = 14$ und $ID = 15$.

Schließlich wird durch den Funktionsaufruf in Zeile 28 bestätigt, dass der Zyklus der vereinigten Masche korrekt bestimmt wurde. Ihr wird die $ID = 5$ zugeordnet.

```
ID | ID_Kanten
5  | -7, 9, 14, -13
```

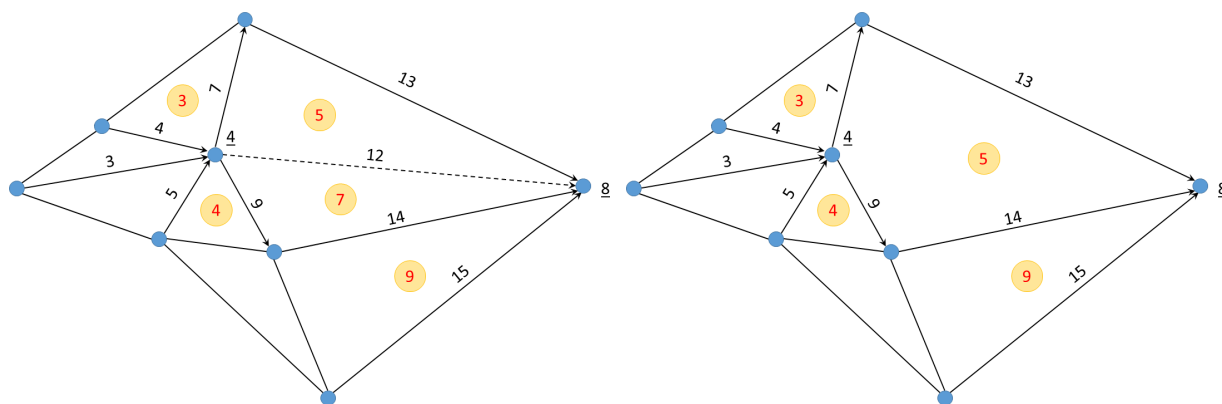
Wird die Kanten-Liste der vereinigten Fläche mit den Listen der zwei Teilflächen vor dem Löschen verglichen, wird schnell deutlich, dass die Listen bei der Verschmelzung umsortiert werden müssen.

Damit dieses recht komplexe Beispiel besser nachvollzogen werden kann, ist in Abbildung der Graph der Zeitlichen Struktur vor und nach dem Löschen der Kante mit allen beteiligten Elementen dargestellt. Die Zahlen entsprechen den IDs aus dem obigen Beispiel, wobei die Kanten-IDs auf den jeweils zugehörigen Elementen stehen, die Knoten-IDs unterstrichen sind und die Flächen-IDs mit gelben Kreisen kenntlich gemacht wurden.

Anschließend an dieses Beispiel werden in den Zeilen 31 und 32 zwei weitere Kanten gelöscht, was dazu führt, dass eine „baumelnde“ Kante entsteht. Diese wird mit dem Plot in Zeile 35 identifiziert und anschließend wird für sie in Zeile 38 eine Konsolenausgabe erzeugt.

```
ID | ... | linker Anrainer | rechter Anrainer | ...
11 | ... | -1 | -1 | ...
```

An der hier präsentierten, gekürzten Fassung der Ausgabe kann leicht überprüft werden, dass die Anrainer der Kante während der Löschvorgänge korrekt aktualisiert wurden.



(a) Basisliniennetzwerk vor dem Löschen der Kante mit der $ID = 12$.

(b) Basisliniennetzwerk nach dem Löschen der Kante mit der $ID = 12$.

Abbildung C.7: Darstellung des Basisliniennetzwerkes mit Angabe aller Elemente, die vom Löschen der Kante mit der $ID = 12$ betroffen sind.

C.9 fuegeKanteEin()

Syntax:

`BS.fuegeKanteEin(ID_A, ID_E, struktur)`

Beschreibung:

Diese Funktion dient dem Einfügen einer Kante zwischen zwei existierenden Knoten in die Zeitliche und Räumliche Struktur. Obwohl in der aktuellen Version lediglich das Einfügen in die Zeitliche Struktur möglich ist, soll im Folgenden der volle zukünftige Funktionsumfang beschrieben werden.

In der Zeitlichen Struktur können Kanten zwischen beliebigen Knoten eingefügt werden. Teilt die Kante eine existierende Masche, so wird diese in zwei Teilflächen zerlegt. Bei allen anderen Einfügeoperationen werden die Kanten keiner Fläche zugeordnet. Da die Operation mit dem Hinzufügen eines Interferogramms zum Bildstapel gleichzusetzen ist, müssen bei aufgebauter Räumlichen Struktur Knoten-, Kanten- und Flächenobjekte erzeugt werden, die mit den existierenden Objekten lageidentisch sind. Hier muss weiterhin darauf geachtet werden, dass die Kante an die richtige Position in der Liste eingefügt wird, damit die chronologische Sortierung des Bildstapels erhalten bleibt.

Das Einfügen einer Kante in die Räumliche Struktur ist nur möglich, wenn eine existierende Fläche dadurch geteilt wird. Hier sind keine Überschneidungen zulässig. Außerdem kommt das Einfügen einer Kante in einem Interferogramm dem Erstellen von lagegleichen Kanten in den restlichen Bildern gleich. Somit wird sichergestellt, dass die Vermaschung in allen Aufnahmen identisch ist.

In einem weiteren Schritt müssen alle Referenzstrukturen aktualisiert werden, um das neue Element ordnungsgemäß in den Bildstapel einzugliedern.

- `BS.fuegeKanteEin(ID_A, ID_E, struktur)` fügt eine Kante zwischen dem Startknoten mit der ID `ID_A` und dem Endknoten mit der ID `ID_E` in die gewünschte `struktur` ein.

Achtung: Das Einfügen von Kanten in die Räumliche Struktur ist momentan noch nicht möglich.

Hinweis: Das Einfügen von Kanten in die Zeitliche Struktur entspricht dem Einfügen eines Interferogramms in den Bildstapel. Entsprechend müssen für diese D-InSAR-Aufnahme neue Objekte in der Räumlichen Struktur erzeugt werden, sofern diese bereits erstellt wurde. Allerdings ist dies in der aktuellen Programmversion noch nicht möglich. Daher sollten die Kanten vor dem Aufbau der Räumlichen Struktur eingefügt werden.

Eingabeparameter:

BS	...	Bildstapel	Referenz auf eine Instanz der Klasse <i>Bildstapel</i> .
ID_A	...	[1x1] double	ID des Anfangsknotens der Kante.
ID_E	...	[1x1] double	ID des Endknotens der Kante.
struktur	...	string	Der String gibt an, ob die Kante in die Zeitlichen (<code>struktur='zeit'</code>) oder in die Räumlichen Struktur (<code>struktur='raum'</code>) eingefügt werden soll.

Beispiel:

Das folgende Beispiel in Listing C.9 schließt gedanklich an das Beispiel zur Funktion `BS.loescheKante()` an (vgl. Funktionsbeschreibung C.8). Dort wurde die Kante mit der `ID = 12` aus der Zeitlichen Struktur entfernt und die Auswirkungen wurden ausführlich dokumentiert. Hier wird nun die Kante wieder eingefügt und gezeigt, dass der Ursprungszustand wieder erreicht wird.

Listing C.9: Einfügen von Kanten in die Zeitliche Struktur.

```

1  % Aufbau der Zeitlichen Struktur
2  BS = Bildstapel(data_dinsar, data_koh, baseline_info, referenz);

4  % Plot des Basisliniennetzwerkes vor der Bearbeitung
5  BS.plot('zeit');

7  % Ausgeben der Zyklen der Anrainer der Kante 12 (Informationen aus dem Plot)
8  BS.showInfo(5, 'f', 'zeit');
9  BS.showInfo(7, 'f', 'zeit');

11 % Loeschen der Kante
12 BS.loescheKante(12, 'zeit');

14 % Plot des Basisliniennetzwerkes nach dem Loeschen
15 BS.plot('zeit');

17 % Ausgeben des Zyklus der vereinigten Masche
18 BS.showInfo(5, 'f', 'zeit');

20 % Wiedereinfuegen der Kante mit der ID=12
21 BS.fuegeKanteEin(4, 8, 'zeit');

23 % Plot des Basisliniennetzwerkes nach dem Einfuegen
24 BS.plot('zeit');

26 % Ausgeben der Zyklen der Anrainer der "neuen" Kante 12 (Informationen aus dem Plot)
27 BS.showInfo(5, 'f', 'zeit');
28 BS.showInfo(7, 'f', 'zeit');
```

In den Zeilen 2–9 werden nacheinander die Zeitliche Struktur aufgebaut, anschließend die Informationen zu den Anrainern der Kante abgefragt, die gelöscht werden soll. Diese wird in Zeile 12 entfernt und in den Zeilen 15–18 werden die Informationen zur neu entstandenen Masche bestimmt.

Nun wird schließlich die Kanten in Zeile 21 wieder eingefügt. Im Laufe dieses Vorgangs werden zwei Dialoge geöffnet. Hier können nun sowohl das differentielle SAR-Interferogramm als auch das Kohärenzbild ausgewählt werden, die der Kante zugeordnet sind. In Abbildung C.8 ist einer der Auswahldialoge beispielhaft dargestellt. In der linken oberen Ecke, die mit einem roten Kasten markiert ist, wird das Produkt angegeben, welches ausgesucht werden muss.

Um zu bestätigen, dass der Ursprungszustand wieder erreicht wurde, wird in Zeile 24 das Basisliniennetzwerk dargestellt. Wie der Abbildung C.9 entnommen werden kann, entsprechen die Kanten-Informationen im

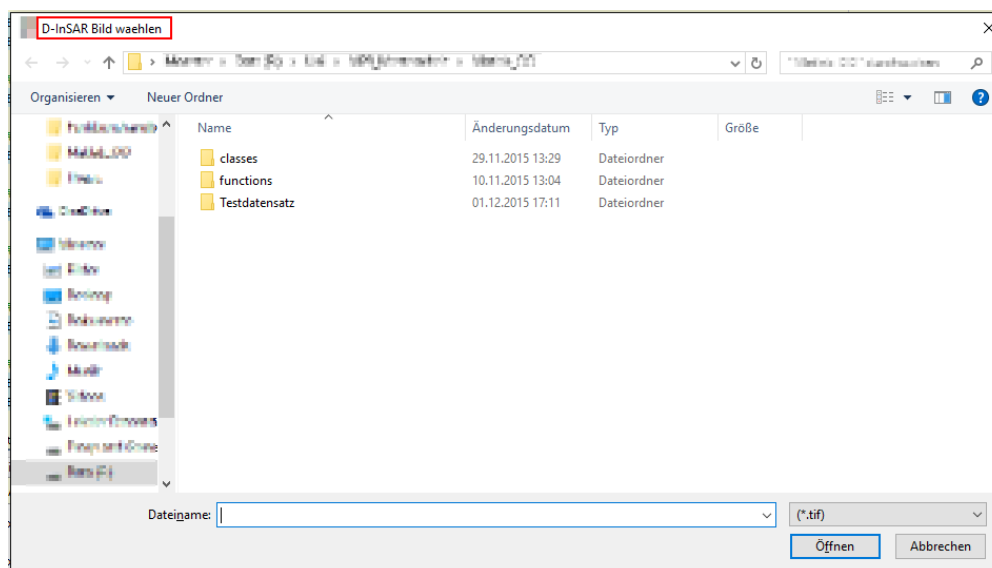


Abbildung C.8: Auswahldialog beim Einfügen von Kanten in die Zeitliche Struktur.

Plot exakt denen aus Abbildung C.6, das den Anfangszustand dokumentiert.

Abschließend werden die Konsolenausgaben für die neu entstandenen Flächen mit den IDs 5 und 7 in den Zeilen 27 und 28 erzeugt. Diese werden den Ausgaben der ursprünglichen Maschen mit den IDs 5 und 7 gegenübergestellt:

Vorher:		Nachher:	
ID	ID_Kanten	ID	ID_Kanten
5	-7, 12, -13	5	-7, 12, -13
7	9, -12, 14	7	9, 14, -12

Es ist zu erkennen, dass die Zyklen für die Fläche mit der $ID = 5$ exakt übereinstimmen. Bei der anderen Fläche hat sich lediglich die Reihenfolge der Kanten geändert, die jedoch auch in der neuen Anordnung einen positiv orientierten Zyklus bilden. Dies kann an Abbildung C.7 leicht nachvollzogen werden.

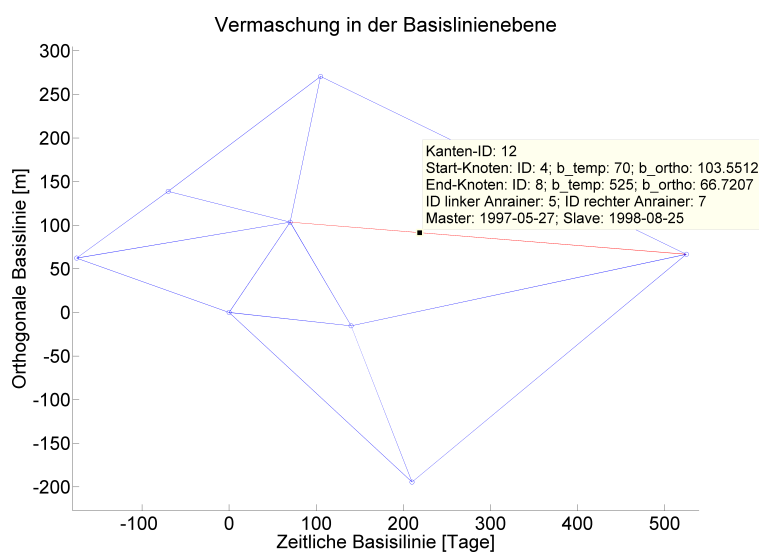


Abbildung C.9: Plot der Zeitlichen Struktur nach Rückeinfügen der Kante mit der $ID = 12$.

Abbildungsverzeichnis

1.1	Überblick über europäische SAR-Missionen.	3
1.2	Differntielles Phasenbild	4
2.1	Einfluss verschiedener Dekorrelationseffekte auf interferometrische Bilddaten	12
2.2	Basisliniennetzwerk eines Single-Master-Stapel	14
2.3	Basisliniennetzwerk eines SBAS-Bildstapels	15
2.4	Basisliniennetzwerk eines EMCF-Bildstapels	16
2.5	Übersicht des Aufnahmegebiets	18
2.6	Struktur des gegebenen Datenstapels in der Basislinienebene	19
3.1	Dreieck einer räumlichen Triangulation auf den kohärenten Pixeln	23
3.2	Zusammenhang zwischen Phasengradienten und dem Bildstapel-Netzwerk	25
3.3	Darstellung der Graphenstruktur für den 3D-Fall des Edgelist-Algorithmus	28
4.1	UML-Diagramm der modellierten Datenstruktur	33
4.2	Konsequenzen des Löschens einer Kante	35
4.3	Konsequenzen des Einfügens einer Kante	35
4.4	UML-Diagramm der Klasse <i>Knotenliste_zeit</i>	36
4.5	UML-Diagramm der Klasse <i>Knoten_zeit</i>	37
4.6	UML-Diagramm der Klasse <i>Kantenliste_zeit</i>	37
4.7	UML-Diagramm der Klasse <i>Kante_zeit</i>	38
4.8	UML-Diagramm der Klasse <i>Flächenliste_zeit</i>	39
4.9	UML-Diagramm der Klasse <i>Fläche_zeit</i>	39
4.10	Speicherung der Kanten IDs für positiv orientierte Flächen.	40
4.11	Sonderfall bei der Zyklensuche in der Zeitlichen Struktur.	41
4.12	Löschen und Einfügen einer Kante in die Räumliche Struktur	42
4.13	UML-Diagramm der Klasse <i>Knotenliste_raum</i>	42
4.14	UML-Diagramm der Klasse <i>Knoten_raum</i>	43
4.15	UML-Diagramm der Klasse <i>Kantenliste_raum</i>	45

4.16	UML-Diagramm der Klasse <i>Kante_raum</i>	46
4.17	UML-Diagramm der Klasse <i>Flächenliste_raum</i>	47
4.18	UML-Diagramm der Klasse <i>Fläche_raum</i>	47
4.19	UML-Diagramm der Klasse <i>Bildstapel</i>	48
5.1	Lage des Testdatensatzes im Aufnahmegebiet.	52
5.2	Zeitliches und Räumliches Netzwerk des Testdatensatzes	53
5.3	Gegenüberstellung von Testdaten mit niedriger und hoher Kohärenz	54
C.1	Zusammenhang zwischen einer Kante im Basisliniennetzwerk und einem Interferogramm	x
C.2	Nutzungsmöglichkeiten der Scatter-Funktion	xi
C.3	Informationsabfrage im Plot des Basisliniennetzwerkes.	xiii
C.4	Informationsabfrage in der Vermaschung der Knoten in einem Interferogramm.	xiv
C.5	Interaktiver Plot mit Informationsabruf zur Kante mit der <i>ID = 12</i>	xix
C.6	Interaktiver Plot mit Informationsabruf zu den Kante mit der <i>ID = 14</i> und <i>ID = 15</i>	xx
C.7	Beteiligte Elemente beim Löschen einer Kante	xxi
C.8	Auswahldialog beim Einfügen von Kanten in die Zeitliche Struktur.	xxiii
C.9	Plot der Zeitlichen Struktur nach Rückeinfügen der Kante mit der <i>ID = 12</i>	xxiii

Tabellenverzeichnis

2.1	Beschreibung des gegebenen Datenstapels	19
5.1	Beschreibung des Testdatensatzes	51
5.2	Funktionsübersicht	55

Literaturverzeichnis

- BERARDINO, P., G. FORNARO, R. LANARI UND E. SANSOSTI (2002): *A New algorithm for Surface Deformation Monitoring Based on Small Baseline Differential SAR Interferograms*. IEEE Transactions on Geoscience and Remote Sensing 40(11), S. 2375–2383.
- BERN, M. UND D. EPPSTEIN (1992): *Mesh Generation And Optimal Triangulation*. In: *Computing in Euclidean Geometry*, Band 1., S. 23–90. World Scientific, Singapur.
- CASU, F. (2009): *The Small BAseline Subset technique: performance assessment and new developments for surface deformation analysis of very extended areas*. Dissertation, Universität Cagliari, Cagliari, Italien.
- CASU, F., M. MANZO UND R. LANARI (2006): *A quantitative assessment of the SBAS algorithm performance for surface deformation retrieval from DInSAR data*. Remote Sensing of Environment 102, S. 195–210.
- CHEN, C. W. (2001): *Statistical-Cost Network-Flow Approaches to Two-Dimensional Phase Unwrapping for Radar Interferometry*. Dissertation, Stanford University, Stanford, Kalifornien.
- CHEN, C. W. UND H. A. ZEBKER (2002): *Phase Unwrapping for Large SAR Interferograms: Statistical Segmentation and Generalized Network Models*. IEEE Transactions on Geoscience and Remote Sensing 40(8), S. 1709–1719.
- COSTANTINI, M. (1997): *A Phase Unwrapping Method Based on Network Programming*. In: *Proceedings of the Fringe 96 Workshop on ERS SAR Interferometry: 30. September–2. Oktober 1996*, Band ESA SP-406, S. 261–272. European Space Agency, Zürich, Schweiz.
- COSTANTINI, M. UND P. ROSEN (1999): *A Generalized Phase Unwrapping Approach for Sparse Data*. In: *Geoscience and Remote Sensing Symposium, 1999. IGARSS'99 Proceedings. IEEE 1999 International*, Band 1, S. 267–269. IEEE, Hamburg.
- DE SOUSA, J. J. M. (2009): *Potential of Integrating PSI Methodologies in the Detection of Surface Deformation*. Dissertation, Universität Porto, Porto, Portugal.
- DINAS, S. UND J. M. BAÑÓN (2014): *A Review on Delaunay Triangulation with Application on Computer Vision*. International Journal of Computer Science and Engineering 3(2), S. 9–18.
- EUROPEAN SPACE AGENCY (ESA) (2015): *NEST - Next ESA SAR Toolbox*. Online unter: <https://earth.esa.int/web/nest/home>. Besucht am 16. Oktober 2015.
- FERRETTI, A., C. PRATI UND F. ROCCA (2001): *Permanent Scatterer in SAR Interferometry*. IEEE Transactions on Geoscience and Remote Sensing 39(1), S. 8–20.
- HALSIG, S. (2012): *Integrierte Modellierung von InSAR- und Nivellementmessungen zur Detektion großräumiger Deformationen der Erdoberfläche*. Masterarbeit, Institut für Geodäsie und Geoinformation der Universität Bonn, Bonn.
- HANSEN, R. F. (2001): *Radar interferometry: Data interpretation and error analysis*. Dissertation, Technische Universität Delft, Delft, Niederlande.
- HOOPER, A. UND H. A. ZEBKER (2007): *Phase unwrapping in three dimensions with application to InSAR time series*. Journal of the Optical Society of America A 24(9), S. 2737–2747.

- HOOPER, A., H. ZEBKER, P. SEGALL UND B. KAMPES (2004): *A new method for measuring deformation on volcanos and other natural terrains using InSAR persistent scatterers*. Geophysical Research Letters 31(L23611).
- HOOPER, A. J. (2006): *Persistent Scatterer Radar Interferometry for Crustal Deformation Studies and Modeling of Volcanic Deformation*. Dissertation, Stanford University, Stanford, Kalifornien.
- JOSTES, A. (2012): *Effiziente Realisierung der bedingten Delaunay Triangulierung in Matlab und C*. Wissenschaftliche Arbeit, Universität Ulm, Ulm.
- KAMPES, B. M. (2005): *Displacement Parameter Estimation using Permanent Scatterer Interferometry*. Dissertation, Technische Universität Delft, Delft, Niederlande.
- KAMPES, B. M. UND N. ADAM (2005): *The STUN Algorithm for Persistent Scatterer Interferometry*. In: *Proceedings of the Fringe 2005 Workshop on ERS SAR Interferometry: 28. November – 2. Dezember 2005*, Band ESA SP-610. European Space Agency, Frascati, Italien.
- KLAUSING, H. UND W. HOLPP (2000): *Radar mit realer und synthetischer Apertur: Konzeption und Realisierung*. Oldenbourg Wissenschaftsverlag, München und Wien, Österreich.
- KNABE, S. (2004): *Erfassung der räumlichen und zeitlichen Veränderung von Bodenfeuchtemustern in semiariden Gebieten mittels ERS-1/2 InSAR Zeitreihen*. Dissertation, Technische Universität Wien, Wien, Österreich.
- LEE, J.-S., K. W. HOPPEL, S. A. MANGO UND A. R. MILLER (1994): *Intensity and Phase Statistics of Multilook Polarimetric and Interferometric SAR Imagery*. IEEE Transactions on Geoscience and Remote Sensing 32(5), S. 1017–1028.
- MADSEN, S. N. (1989): *Estimating The Doppler Centroid of SAR Data*. IEEE Transactions on Aerospace and Electronic Systems AES-25(2), S. 134–140.
- MADSEN, S. N. UND H. A. ZEBKER (1998): *Imaging Radar Interferometry*. In: *Principles and Applications of Imaging Radar*, Band 2 von *Manual of Remote Sensing*, Kapitel 6. John Wiley & Sons, New York.
- MOLL, A. (2007): *Radarinterferometrische Untersuchungen mit ERS-1/2 auf der Antarktischen Halbinsel*. Dissertation, Universität Bonn, Bonn.
- PAPADIMITRIOU, C. H. UND K. STEIGLITZ (1998): *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, New York, 2. Auflage.
- PEPE, A. UND R. LANARI (2006): *On the Extension of the Minimum Cost Flow Algorithm for Phase Unwrapping of Multitemporal Differential SAR Interferograms*. IEEE Transactions on Geoscience and Remote Sensing 44(9), S. 2374–2383.
- PEPE, A., M. MANUNTA, L. EUILLADES, L. PAGLIA, Y. YANG UND R. LANARI (2011): *New Improvements Of The EMCF Phase Unwrapping Algorithm For Surface Deformation Analysis At Full Spatial Resolution Scale*. In: *Proceedings of Fringe 2011: 19.–23. September 2011*, Band ESA SP-697, S. 19–24. European Space Agency, Frascati, Italien.
- RICHARDS, M. A. (2006): *A Beginner's Guide to Interferometric SAR Concepts and Signal Processing*. IEEE Aerospace and Electronic Systems Magazine 22(9), S. 5–29.
- ROSEN, P. A., S. HENSLEY, I. R. JOUGHIN, F. K. LI, S. N. MADSEN, E. RODRÍGUEZ UND R. M. GOLDSTEIN (2000): *Synthetic Aperture Radar Interferometry*. Proceedings of the IEEE 88(3), S. 333–382.

- SCHÄFER, M. (2012): *Atmosphäre als Phasenbestandteil der differentiellen Radarinterferometrie und ihr Einfluss auf die Messung von Höhenänderungen*. Dissertation, Fakultät für Energie- und Wirtschaftswissenschaften der Technischen Universität Clausthal, Clausthal.
- SEIFERT, P. UND M. ZINK (1993): *Synthetisch-Aperture Radar: Technik und Anwendung*. Physik in unserer Zeit S. 24–32.
- SHANKER, A. P. (2010): *Persistent Scatterer Interferometry in Natural Terrain*. Dissertation, Stanford University, Stanford, Kalifornien.
- SHANKER, A. P. UND H. A. ZEBKER (2010): *Edgelist phase unwrapping algorithm for time series InSAR analysis*. Journal of the Optical Society of America A 27(3), S. 605–612.
- SÖRGEL, U. (2006): *Stand der Technik und Trends der Radarfernerkundung*. In: *Festschrift 125 Jahre Geodäsie und Geoinformatik. Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover*.
- TOUZI, R., A. LOPES, J. BRUNIQUEL UND P. W. VACHON (1999): *Coherence Estimation for SAR Imagery*. IEEE Transactions on Geoscience and Remote Sensing 37(1), S. 135–149.
- WALTER, D., J. HOFFMANN, N. ADAM UND B. KAMPES (2004): *Radarfernerkundung zur Erfassung bergbaubedingter Senkungen*. 5. Geokinematischer Tag Nr. 2, S. S. 228–239.
- WALZOG, S. (2014): *3-D Phase Unwrapping Algorithmen zur Lösung der Phasenmehrdeutigkeiten in D-InSAR Stapeln*. Masterarbeit, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn.
- WONG, F. UND I. G. CUMMING (1996): *A Combined SAR Doppler Centroid Estimation Scheme Based upon Signal Phase*. IEEE Transactions on Geoscience and Remote Sensing 34(3), S. 696–707.
- YANNAM, P. M., P. V. JAYASRI, K. RAMESHBABU UND S. B. CHADA (2012): *Estimation of Doppler Centroid Frequency Using SAR Imaging Geometry for RISAT-2*. International Journal of Engineering and Advanced Technology 2(2), S. 43–51.
- ZEBKER, H. A. UND J. VILLASENOR (1992): *Decorrelation in Interferometric Radar Echoes*. IEEE Transactions on Geoscience and Remote Sensing 30(5), S. 950–959.