

Master Thesis

im Rahmen des
Universitätslehrganges „Geographical Information Science & Systems“
(UNIGIS MSc) am Zentrum für GeoInformatik (Z_GIS)
der Paris Lodron-Universität Salzburg

zum Thema

„Massenprozessierung von Laserscan-Daten“ Erhebung des Streuobstbaumbestandes in Baden- Württemberg

vorgelegt von

Dipl. Ing. (FH) Holger Bischooping
U1332, UNIGIS MSc Jahrgang 2007

Zur Erlangung des Grades
„Master of Science (Geographical Information Science & Systems) – MSc(GIS)“

Gutachter:
Ao. Univ. Prof. Dr. Josef Strobl

Herrsching am Ammersee, 09. November 2009



Es ist ein übliches Verfahren, das Bild von uns selbst in Dinge oder in Funktionen von Dingen zu projizieren.

Heinz von Foerster, Bemerkungen zu einer Epistemologie des Lebendigen

Erklärung

Ich, Holger Bischooping, geboren am 03.12.1971 in Wadern, erkläre, diese Master Thesis ohne fremde Hilfe und ohne Verwendung anderer als der angeführten Quellen angefertigt zu haben, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt wurde.

Alle Ausführungen der Arbeit, die wörtlich oder sinngemäß übernommen wurden, sind entsprechend gekennzeichnet.

Dipl.-Ing. (FH) Holger Bischooping

Kurzfassung

Im Rahmen eines Verbundprojekts wird der aktuelle Streuobstbaumbestand des Landes Baden-Württemberg landesweit und flächendeckend erfasst und mit früheren Daten verglichen. Als Datengrundlage dienen Airborne Laserscan Daten, die zur Erhebung des digitalen Geländemodells des Landes erstellt wurden sowie eine Maske (Polygon-Feature), die nur diejenigen Flächen auszeichnet, auf denen Streuobstbestände vorkommen können. Die Maske basiert auf Informationen des amtlichen Liegenschaftskatasters (ALK) sowie des ATKIS®-Basis-DLM.

Ziel der vorliegenden Arbeit ist die Untersuchung, inwieweit die zugrunde liegenden Daten mit einer für die Aufgabe sehr geringen Auflösung von etwa einem Punkt pro m² für eine Erhebung des Streuobstbaumbestandes geeignet sind. Es wird ein speziell auf die Daten abgestimmter Algorithmus zur Einzelbaumextraktion auf den methodischen Grundlagen bestehender Algorithmen entwickelt. Zur Validierung dient eine aus Orthophotos erstellte visuelle Referenz, mit der die Ergebnisse abgeglichen werden. Eine qualitative Aussage über die Güte des Algorithmus auf Basis der Orthophotos muss jedoch kritisch betrachtet werden, da die Erhebung der Laser Scan Daten und der Orthophotos teilweise mehr als 5 Jahre auseinandergelegen hat. Mit den verfügbaren Validierungsdaten kann somit nur eine Vollständigkeit von 55 % und eine Korrektheit von 64 % nachgewiesen werden.

Weiterhin wird im Rahmen dieser Arbeit untersucht, ob sich die Technologie des Desktop Grid Computing als Werkzeug für die verteilte Massenprozessierung von Fernerkundungsdaten eignet. Zu diesem Zweck wird ein System auf der Basis der frei verfügbaren Plattform BOINC eingerichtet, das die gekachelten Eingangsdaten auf mehreren Arbeitsrechnern eines Netzwerks verteilt und prozessiert. Es konnte gezeigt werden, dass die Verarbeitung datenintensiver Fernerkundungsdaten mit Hilfe einer solchen Plattform möglich ist. Zur Bewertung des Systems wird ein rudimentäres Vergleichssystem auf der Basis eines Einzelrechners mit 8 Prozessoren aufgebaut. Es hat sich gezeigt, dass der Zeitaufwand für die Verteilung datenintensiver Fernerkundungsdaten im Netzwerk einen maßgeblichen Einfluss auf die Gesamtverarbeitungsgeschwindigkeit hat. Als Ergebnis kann festgehalten werden, dass sich der Aufwand zur Einrichtung eines solchen Desktop Grid Computing Systems für Bildanalysen in der Fernerkundung nur bei rechenintensiven Verarbeitungen lohnt.

Abstract

In the framework of a joint research project the actual stock of mixed orchards is assessed for the German province Baden-Württemberg. The research is based on Airborne Laserscan Data with the resolution of about one point per m² (originally acquired for a digital terrain model of Baden-Württemberg), as well as a mask in the form of a polygon feature consisting of potential areas for orchards. This information was extracted from the land survey register and the ATKIS®-Basis-DLM.

The intent of this thesis is to analyse the feasibility of using this kind of data for an acquisition of fruit tree orchards. The resulting fruit trees extracted by the specifically developed algorithm are validated with a visual analysis on the basis of several aerial photos. But this comparison has to be seen critically since there is a time difference of up to five years between the acquisition of the laser data and the aerial photos. By using this validation data the algorithm achieves a completeness of about 55 % and an accuracy of about 64 %.

The second part of this thesis additionally investigates the suitability of Desktop Grid Computing Technology as tool for parallel processing of huge amounts of remote sensing data. For this purpose a system based on the open source platform BOINC is created to distribute and process the tiled data by using desktop computers attached to the network. To evaluate this distributed computing system a simple system is built consisting of only one desktop computer with eight processors. The comparison of these two systems leads to the result that parallel processing of huge data amounts is very time-consuming due to the necessary data transfer within the network. This system is only efficient for either small data or very complex image analyses requiring a long processing time.

Inhaltsverzeichnis

<i>Erklärung</i>	2
<i>Kurzfassung</i>	3
<i>Abstract</i>	4
<i>Inhaltsverzeichnis</i>	5
<i>Abbildungsverzeichnis</i>	8
<i>Tabellenverzeichnis</i>	11
<i>Abkürzungsverzeichnis</i>	12
1 Einleitung	1
1.1 Motivation	1
1.2 Fragestellung und Ziele	3
1.3 Nicht behandelte Themen	4
1.4 Gliederung der Arbeit	5
2 Stand der Technik	7
2.1 Airborn Laserscanning	7
2.1.1 Messverfahren	7
2.1.2 Reflexionsintensität	9
2.2 Bildanalyse zwischen Fernerkundung und Geoinformatik	9
2.2.1 Bildverstehen	9
2.2.2 Digitale Bildverarbeitung in der Fernerkundung	10
2.2.3 Wissen und wissensbasierte Bildanalyse	12
2.3 Ausgewählte Themen der Bildverarbeitung	13
2.3.1 Delaunay-Triangulation	13
2.3.2 Räumliche Interpolationsverfahren	14
2.3.3 Morphologische Mathematik	14
2.3.4 Segmentierung	17
2.3.5 Rasterdatenzerlegung	19
2.3.6 Bewertungsverfahren für Segmentierungs- und Klassifikationsergebnisse	21
2.4 Objektextraktion von Bäumen aus Fernerkundungsdaten	24
2.4.1 Der Begriff „Streuobst“	24
2.4.2 Bisherige Arbeiten zur Baumextraktion	24
2.5 Grundlagen der Parallelen Prozessierung	29
2.5.1 Parallelen Prozessierung in der Fernerkundung	29
2.5.2 Merkmale paralleler und verteilter Systeme	30
2.5.3 Cluster-Computing	31
2.5.4 Grid-Computing	33
2.5.5 Desktop Grid-Computing	35
2.6 Das System BOINC	35
2.6.1 Die Idee	35
2.6.2 Die Architektur von BOINC	36
2.6.3 Wrapper-Konfiguration	39

2.6.4	Job-Verteilverfahren	39
3	<i>Material und Untersuchungsgebiet</i>	40
3.1	Untersuchungsgebiet, Testdaten und Referenzflächen	40
3.1.1	Untersuchungsgebiet	40
3.1.2	Testdaten für die Entwicklung des Algorithmus	41
3.1.3	Referenzflächen für die Validierung	41
3.1.4	Zusätzliche Daten für Entwicklung und Validierung	42
3.2	Laserscan-Daten des Landesvermessungsamts Baden-Württemberg	43
3.2.1	Befliegungskampagne	43
3.2.2	Eckdaten und Parameter der ALS-Datenerfassung	44
3.2.3	Genauigkeit der Laserscandaten	45
3.2.4	Nachbearbeitung der Rohdaten	45
3.3	Abgrenzung des Kontext	47
3.3.1	Kontextmodell	47
3.3.2	Normalisiertes Digitales Oberflächenmodell	48
3.3.3	Automatisierte Liegenschaftskarte (ALK)	48
3.3.4	ATKIS Basis DLM	48
3.3.5	Aufbau der Maske	49
3.3.6	Qualität der Maske	51
4	<i>Methoden und Werkzeuge</i>	52
4.1	Methoden der Entwicklung des Algorithmus zur Einzelbaumextraktion	52
4.1.1	Theorieansatz	52
4.1.2	Vorgehensweise bei der Entwicklung des Algorithmus	54
4.1.3	Werkzeuge der Entwicklung	55
4.1.4	Validierung des Einzelbaumextraktionsalgorithmus	55
4.2	Methoden der Entwicklung des Massenprozessierungssystems	55
4.2.1	Methodische Vorgehensweise bei der Recherche	55
4.2.2	Katalogisierung der Eingangsdaten	56
4.2.3	Aufbau des Massenprozessierungssystems	57
4.2.4	Bewertung des Massenprozessierungssystems	57
5	<i>Ergebnis des Algorithmus zur Einzelbaumextraktion</i>	58
5.1	Algorithmus zur Einzelbaumextraktion	58
5.1.1	Ein- und Ausgangsdaten des Algorithmus	58
5.1.2	Konfigurationsparameter des Algorithmus	59
5.1.3	Ablauf des Algorithmus	60
5.2	Validierung der Ergebnisse	67
5.2.1	Performance	67
5.2.2	Genauigkeitsanalyse	68
6	<i>Ergebnis des Massenprozessierungssystems</i>	70
6.1	Aufbau des Massenprozessierungssystems	70
6.2	Das Subsystem „Katalogisierung“	70
6.2.1	Struktur der Eingangsdaten	70
6.2.2	Optimierung von Überlappungsbereich und Prozesskachelgröße	71
6.2.3	Von den Eingangsdaten zu den Prozessdaten	73
6.2.4	Datenbankschema des Katalogs	75
6.2.5	Ablauf der Reorganisation	76

6.3	Das Subsystem „Verteiltes Rechnen“	77
6.3.1	Systemaufbau	77
6.3.2	Systemkonfiguration	78
6.4	Bewertung des Massenprozessierungssystems	79
6.4.1	Vergleichssystem	79
6.4.2	Ergebnis des Vergleichs	80
7	<i>Diskussion und Ausblick</i>	81
7.1	Diskussion	81
7.2	Ausblick	83
8	<i>Literatur und Quellen</i>	85
8.1	Literaturverzeichnis	85
8.2	Softwareverzeichnis	89

Abbildungsverzeichnis

Abbildung 1: Methodische Vorgehensweise in den beiden Teilen der Arbeit	6
Abbildung 2: Gegenüberstellung von first pulse/last pulse und full waveform (Quelle: GeoLas Consulting)	8
Abbildung 3: Schematische Darstellung des DOM (grün) und DGM (brau); (Quelle:Topsan).....	8
Abbildung 4: Unterschiedliche Ausgangslage von Buttom-Up- und Top-Down-Methoden	9
Abbildung 5: Beziehung zwischen Auflösung und der Größe realweltlicher Objekte...	11
Abbildung 6: GIS-Daten als zusätzliches Ausgangs- und Ergebnisprodukt der fernerkundlichen Bildanalyse.....	12
Abbildung 7: Delaunay-Triangulation	14
Abbildung 8: Morphologische Basisoperationen Dilatation und Erosion	16
Abbildung 9: Gegenüberstellung der Basisoperationen und der zusammengesetzten Operationen (Soille 2003).....	16
Abbildung 10: Objekt außerhalb des Blocks aber innerhalb des Überlappungsbereichs	20
Abbildung 11: Rasterdatenzerlegung ohne (links) und mit Überlappungsbereich (rechts)	20
Abbildung 12: Konfusionsmatrix.....	22
Abbildung 13: Anzahl der Objekte in Referenz und Extraktionsergebnis.....	23
Abbildung 14: 3-D-Segmentierung auf Full Waveform LIDAR Daten (Reitberger et al. 2009)	26
Abbildung 15: 3-D-Visualisierung des Pollock-Modells; Links: Laubbaums; Rechts: Nadelbaum	27
Abbildung 16: Grid-Infrastruktur mit Schnittstellen zu Sensor- und Datenverarbeitungssystemen.....	34
Abbildung 17: Komponenten der BOINC-Plattform.....	37
Abbildung 18: Screenshot des Boinc-Manager	38
Abbildung 19: Kapselung der Projektanwendung durch die Wrapper-Anwendung	39

Abbildung 20: Kachelung des Untersuchungsgebietes.....	40
Abbildung 21: Auswahl der Referenzflächen	42
Abbildung 22: Verdeutlichung der zeitlichen Differenz von ALS-Daten und Orthophoto (oben) bzw. Google Earth (unten).....	43
Abbildung 23: Skizze der Befliegungskampagne aus (Screenshot aus (Klauser 2008) .	44
Abbildung 24: Arbeitsschritte zur Erstellung des digitalen Geländemodells (vgl. Schleyer 2001)	46
Abbildung 25: Beispiele von Fehlerquellen bei der Generierung des DGM (Klauser 2008)	47
Abbildung 26: Aktivitätsdiagramm der Maskengenerierung.....	49
Abbildung 27: Vereinfachte Darstellung der Arbeitsschritte bei der Generierung der Maske	50
Abbildung 28: Fehler in der generierten Maske	51
Abbildung 29: Eingangs- und Ausgangsdaten des Algorithmus	58
Abbildung 30: Darstellung einer Prozesskachel mit First Pulse-Daten.....	59
Abbildung 31: Detailliertes Aktivitätsdiagramm mit alle Teilschritten.....	60
Abbildung 32: Übetragung der First Pulse Daten auf ein Raster mit der Rasterweite 1 m62	
Abbildung 33: Gesampelte Last Pulse Bodenpunkte mit Lücken vor der Interpolation (links) und danach (rechts).....	62
Abbildung 34: Gesampelte First Pulse Daten mit Lücken vor der Interpolation (links) und danach (rechts)	63
Abbildung 35: Maskiertes und höhengefiltertes nDOM.....	64
Abbildung 36: Binäres nDOM und Distanzbild des invertierten binären nDOM	65
Abbildung 37: Negiertes Distanzbild und Ergebnis der Watershed-Transformation	65
Abbildung 38: Falschfarbendarstellung der Watershed-Transformation.....	66
Abbildung 39: Auszug aus den Referenzflächen mit gepufferten Validierungsdaten und den Bäumen der Objektextraktion	68
Abbildung 40: Prinzipieller Aufbau des Massenprozessierungssystems.....	70

Abbildung 41: Überlappungsbereich löst das Problem der Objektzerschneidung	71
Abbildung 42: Voruntersuchung: Häufigkeitsverteilung des Durchmessers der konvexen Hülle.....	72
Abbildung 43: Maß der effektiven Zeitkomplexität	73
Abbildung 44: Zoom vom Untersuchungsgebiet in die Eingangsdaten und von dort in die Prozessdatenebene.....	74
Abbildung 45: Adressierung der Nachbarzellen im regulären Gitter	74
Abbildung 46: Datenablage von Eingangs- und Prozessdaten	75
Abbildung 47: ER-Diagramm des Katalogdatenbankschemas	76
Abbildung 48: Vereinfachter Ablauf der Reorganisation	77
Abbildung 49: Systemaufbau.....	78
Abbildung 50: Verriegelungsmechanismus des Vergleichssystems.....	79
Abbildung 51: Häufigkeitsverteilung der Laufzeiten im BOINC-System und im Vergleichssystem	80

Tabellenverzeichnis

Tabelle 1: Morphologischer Ansatz nach Andersen et al.	28
Tabelle 2: Watershedtransformation im Maßstabsraum nach Straub	28
Tabelle 3: NATSCAN-Projekt von Diedershagen et al.	28
Tabelle 4: HIGHSCAN nach Hyypä et al.	29
Tabelle 5: Hill-Climbing nach Persson	29
Tabelle 6: Objektmodellbeschreibung	52
Tabelle 7: Nicht berücksichtigte Modellannahmen	53
Tabelle 8: Objektmerkmale.....	53
Tabelle 9: Vollständigkeit und Korrektheit der Objektklasse „kleine Bäume“	69
Tabelle 10: Vollständigkeit und Korrektheit der Objektklasse „große Bäume“	69

Abkürzungsverzeichnis

Abkürzung	Beschreibung
ALS	Airborne Laser Scanning
API	Application Programming Interface (Programmierschnittstelle)
ASCII	American Standard Code for Information Interchange
ATKIS	Amtliches Topographisch-Kartographisches Informationssystem
AvD	Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland
BOINC	Berkeley Open Infrastructure for Network Computing
BMBF	Bundesministerium für Bildung und Forschung
CHM	Canopy Height Modell (Baumkronenmodell; Eine spezielle Form des normalisierten DOM)
DGM	Digitales Geländemodell (engl. Digital Terrain Model, DTM)
DHM	Digitales Höhenmodell (engl. Digital Elevation Model, DEM)
DOM	Digitales Oberflächenmodell (engl. Digital Surface Model, DSM)
ESRI	Environmental Systems Research Institute
ETRS89	Europäisches Terrestrisches Referenzsystem 1989
GDAL	Geospatial Data Abstraction Library
GDI	Geodateninfrastruktur
GIS	Geo-Informationssystem
GNU	GNU steht für "GNU is not Unix"
GPL	GNU General Public License
GRS80	Geodätisches Referenzsystem 1980 , engl. Geodetic Reference System 1980
HP	Hewlett-Packard Company
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
KMU	Kleine und mittlere Unternehmen
Lidar	Light detection and ranging
MLR	Baden-Württembergische Ministerium für Ernährung und Ländlichen Raum
NASA	National Aeronautics and Space Administration
NAS	Datenformat „Normbasierte Austauschschnittstelle“ ist eine Datenschnittstelle zum Austausch von Geoinformationen
OGC	Open Geospatial Consortium
OGF	Open Grid Forum
OGR	GDAL/OGR Simple Feature Library
OGSA	Open Grid Service Architecture
SHAPE	Dateiformat (Endung *.shp) der Firma ESRI
TCP/IP	Transmission Control Protocol / Internet Protocol
WGS84	World Geodetic System 1984
VM	Virtual Machine

1 Einleitung

1.1 Motivation

Das Problem

Der Obstbau in Baden-Württemberg besitzt eine lange Tradition. Erste urkundliche Belege über "Baumgärten" und "Baumpflanzungen" gehen bis in das achte Jahrhundert zurück.

Die aktuelle Situation des Lebensraums „Streuobstwiese“ ist durch mehrere Entwicklungen bedroht. Die Streuobstwiesen beschränken sich zumeist auf Grenzertragsstandorte, aus denen sich die Landwirtschaft zunehmend zurückzieht, so dass sowohl die extensive Grünlandbewirtschaftung, als auch die Ersatzpflanzung und Pflege der Bäume zur Obstproduktion für die Zukunft großflächig nicht mehr gewährleistet ist. Eine Pflege ist aufgrund der klein strukturierten Besitzverhältnisse sehr aufwändig (Schmieder 2008).

Die Folge ist oft die Aufgabe dieser Nutzungsart zugunsten mehr oder weniger frequenter Mulchverfahren, Verbrachung, Aufforstung oder die Bebauung der attraktiven und für die Landwirtschaft „unnütz“ gewordenen sonnigen Hanglagen, was mit einem Rückgang der Grünlandarten als wichtige Grundlage für die faunistische Vielfalt einhergeht. Der Rückgang der Streuobstwiesenflächen mit ihrer kaum zu erfassenden Artenvielfalt geht einher mit dem Verlust wertvoller genetischer Ressourcen.

Der Streuobstbau stellt nach wie vor sowohl hinsichtlich der bewirtschafteten Fläche, als auch der durchschnittlichen Erntemengen die dominierende Form der Obstproduktion in Deutschland und insbesondere in Baden-Württemberg dar. Nach den Ergebnissen der repräsentativen Streuobsterhebung von 1990, welche die jüngste Erhebung des Streuobstbaus in Baden-Württemberg darstellt, ist die Zahl der Streuobstbäume im Land auf mehr als 11 Millionen zu beziffern. Nach Angaben des Ministeriums für Ernährung und Ländlichen Raum (Stächele 2003) verfügt Baden-Württemberg mit diesen auf schätzungsweise 180.000 Hektar verteilten Streuobstbäumen über die bundesweit umfangreichsten Streuobstbestände. Der aktuelle Pflegenotstand der Obstbaumbestände lässt jedoch vermuten, dass innerhalb der nächsten zehn bis zwanzig Jahre vielerorts ein flächiges Zusammenbrechen der Altbäume zu erwarten ist.

Exakte Erhebungen über die aktuelle Zahl der hochstämmigen Obstbäume existieren nicht. Die letzte flächendeckende Streuobsterhebung in Baden-Württemberg wurde im Jahr 1965, die letzte repräsentative im Jahr 1990 durchgeführt.

Der Zusammenhang zwischen dem Struktureichtum einer Landschaft und der Artenvielfalt ist durch zahlreiche Untersuchungen belegt (Mühlenberg 1997). Der Beitrag der Streuobstwiesen zur Biodiversität und zur Lebensraumqualität für den Menschen erfordern innovative Schutz- und Bewirtschaftungskonzepte, die sowohl Naturschutzgesichtspunkten Rechnung tragen, als auch den sozioökonomischen Rahmenbedingungen angepasst sind. Voraussetzung für solch innovative Konzepte ist die Erfassung des Status quo der Größe der aktuellen Anbauflächen und deren Zustandsbewertung in Bezug auf die Altersstruktur und den Pflegezustand der Obstbäume, welche auch für die Fruchtsaftindustrie eine Einschätzung der mittelfristigen Perspektiven des Angebots erlauben würde.

Das Projekt

Im Rahmen eines Verbundprojekts, an dem mehrere Institute sowie ein mittelständisches Unternehmen aus dem Bereich Fernerkundung beteiligt sind, wird das Ziel verfolgt, den Streuobstbaumbestand des Landes in Baden-Württemberg landesweit und flächendeckend zu erfassen.

Diese quantitative Erhebung wird mittels Fernerkundungsverfahren auf der Basis aktueller, landesweit verfügbarer Laserscan-Daten (ALS-Daten) und Daten des digitalen Landschaftsmodells des Landesvermessungsamtes Baden-Württemberg durchgeführt.

Die dabei zugrunde liegenden ALS-Daten wurden speziell für die Erstellung des landesweiten digitalen Geländemodells erfasst (Schleyer 2001). Daher gilt es zunächst zu klären, inwieweit sich diese Daten für eine Erhebung des aktuellen Streuobstbaumbestandes eignen. Die wesentlichen Ziele des Verbundprojekts sind:

- Der Einsatz von Fernerkundungs- und GIS-Verfahren zu quantitative Erfassung des aktuellen Baumbestandes und der Streuobstwiesenflächen Baden-Württembergs
- Eine Fehlerabschätzung der Ergebnisse mittels Geländekartierungen
- Die Bilanzierung der Veränderungen im Vergleich mit den Daten der landesweiten Kartierung von 1965.

1.2 Fragestellung und Ziele

Die Ausgangssituation dieser Arbeit besteht in einer typischen Problematik der Geoinformatik: Für eine gegebene Fragestellung stehen häufig nur unzureichend geeigneten Daten zur Verfügung.

Im vorliegenden Fall sollen aus den Laser Scan Daten, die eigens zur Erstellung eines neuen digitalen Geländemodells des Landes Baden-Württemberg erfasst wurden, Streuobstbäume detektiert werden. Diese Daten bergen drei grundsätzliche Schwierigkeiten. Die Informationsdichte (räumliche Auflösung) der Daten liegt in einem Bereich, in dem es schwierig ist, Einzelobjekte der Größe eines Baumes zu extrahieren. Das Fehlen einer klaren Definition des Begriffs „Streuobstbaum“ erschwert diese Problematik zusätzlich. Zudem fanden die Befliegungen der Laser Scan Daten in den vegetationsarmen Wintermonaten statt, so dass nur wenigen Reflexionswerte zur Charakterisierung eines Einzelobjekts „Baum“ zur Verfügung stehen.

Die grundlegende **Fragestellung** der vorliegenden Arbeit besteht also darin, zu untersuchen, inwieweit sich die verfügbaren ALS-Daten, für die Detektion von Einzelbäumen eignen. Als **Ergebnis** dieser Untersuchung wird ein konkreter Algorithmus zur Einzelbaumextraktion entwickelt, der auf der Grundlage verfügbarer Validierungsdaten bewertet wird.

Eine **zweite Fragestellung** beschäftigt sich mit der Verarbeitung der großen Menge an Fernerkundungsdaten, die bei einer solch flächendeckenden Erhebung vorliegen. Es wird untersucht, welche Technologien der Parallelen Verarbeitung sich für die Massenprozessierung von ALS-Daten eignen. Dabei wird das **Ziel** verfolgt, eine konkrete Systemumgebung einzurichten, in der zunächst nur Teile des Gesamtuntersuchungsgebietes prozessiert werden können. Im Vorfeld werden konkrete Anforderungen an ein solches System aus der Sicht eines mittelständischen Unternehmens der fernerkundlichen Bilddatenverarbeitung definiert.

Beabsichtigtes Publikum

Zunächst wendet sich diese Arbeit an all diejenigen, die sich mit Methoden der Einzelbaumextraktion auf der Grundlage von ALS-Daten befassen. Dabei sollen insbesondere die Projektbeteiligten der unterschiedlichen Disziplinen des übergeordneten Verbundprojekts angesprochen werden. Aus diesem Grund werden die wichtigsten Grundlagen zum Verständnis des Algorithmus in Kapitel 2.3 erläutert.

An zweiter Stelle wendet sich die Arbeit an ein Fachpublikum aus dem Umfeld Fernerkundung und Geoinformatik, das an konkreten Umsetzungen einer parallelen Verarbeitung großer Datenmengen interessiert ist.

1.3 Nicht behandelte Themen

Grundlagen

Im Rahmen dieser Arbeit wird auf eine detaillierte Einführung in die Themenkomplexe Geoinformationssysteme und Fernerkundung verzichtet. Es wird auf entsprechende Standardwerke verwiesen (Longley 2007, Lillesand, Kiefer & Chipman 2008, Albers 2007).

Software

Die im Rahmen dieser Arbeit verwendeten Softwaresysteme, deren Installation sowie einzelne Konfigurationen werden nicht im Detail diskutiert. Insbesondere wird auf eine ausführliche Beschreibung der Installation einer virtuellen Maschine (VMware Server) auf einem Host-System sowie der Installation und Konfiguration der verwendeten virtuellen Maschine aus dem Projekt BOINC verzichtet (VMware Server 2009, University of California 2009).

Klassifikation

Als Ergebnis liefert der implementierte Einzelbaumextraktionsalgorithmus Baumpunkte und Baumkonturen. Diese werden wertneutral an die Projektpartner ausgeliefert. Es erfolgt keine Klassifikation, beispielsweise auf der Grundlage einer Clusteranalyse oder eines Fuzzy-Klassifikators. Weiterhin erfolgt keine Generalisierung der Ergebnisdaten wie beispielsweise ein Buffering der gefundenen Konturen zur optischen Glättung. Die anwendungsbezogene Interpretation der Daten ist nicht Teil dieser Arbeit sondern wird im Rahmen übergeordneten Verbundprojekts von den beteiligten Projektpartnern durchgeführt.

Vollständige Prozessierung

Die eigentliche Prozessierung des gesamten Untersuchungsgebietes Baden-Württemberg ist ebenfalls nicht Teil der vorliegenden Arbeit. Und somit ist auch eine Auswertung der Gesamtergebnisse der Erhebung des Streuobstbaumbestandes nicht Inhalt dieser Arbeit. Es werden lediglich Untersuchungen und Ergebnisse anhand von Stichprobengebieten gegeben.

1.4 Gliederung der Arbeit

In **Kapitel 1** wird an das Thema herangeführt und das Verbundprojekt „Einsatz von Fernerkundungs- und GIS-Verfahren zur Erfassung von Streuobstwiesenflächen und Baumbestandszahlen in Baden-Württemberg“ vorgestellt. Anschließend werden Fragestellung, Ziele und die erwarteten Ergebnisse erörtert. Es folgt eine Adressierung des beabsichtigten Publikums sowie eine inhaltliche Abgrenzung nicht behandelter Themen.

In **Kapitel 2** wird der Stand der Technik zusammengefasst. Anhand ausgewählter Hintergrundinformationen wird an das Thema der Einzelbaumextraktion herangeführt. Zunächst wird das Verfahren des Airborne Laserscanning kurz beschrieben. Im folgenden Unterkapitel „Bildanalyse zwischen Fernerkundung und Geoinformatik“ wird die Entwicklung der digitalen Bildverarbeitung hin zur objektorientierten Bildanalyse beschrieben. Anschließend werden einzelne, für diese Arbeit relevante, Teilgebiete der Bildverarbeitung wie Interpolationsverfahren, Segmentierung und die mathematische Morphologie erörtert. Es folgt eine Zusammenfassung bisheriger Arbeiten zum Thema Objektextraktion von Bäumen aus Fernerkundungsdaten, wobei hier nur Studien dargestellt werden, die ebenfalls auf ALS-Daten basieren. Kapitel 2 schließt mit einer allgemeinen Einführung in die Grundlagen der Parallelen Prozessierung, mit der Absicht, das in der vorliegenden Arbeit realisierte Massenprozessierungssystem in einen technologischen Kontext zu stellen.

Die Beschreibung des Untersuchungsgebiets leitet das **Kapitel 3** „Material und Untersuchungsgebiet“ ein. Es folgt eine Beschreibung der Testdaten, die im Rahmen der Entwicklung des Algorithmus verwendet werden, sowie der Referenzflächen, die der Validierung des Algorithmus zugrunde liegen. Nach dieser terminologischen Trennung der Begriffe Testdaten und Referenzflächen wird im Detail auf die der Arbeit zugrunde liegenden ALS-Daten und der Masken des Kontextmodells eingegangen.

Kapitel 4 fasst die verwendeten Methoden und Werkzeuge zusammen. Dabei wird wiederum zwischen den beiden Teilen „*Algorithmusentwicklung*“ auf „*Aufbau eines Massenprozessierungssystems*“ unterschieden. Abbildung 1 skizziert diese Trennung der beiden Teile und veranschaulicht stark vereinfacht die jeweilige Vorgehensweise. Ausgangspunkt und Ergebnisse beider Teile sind die Datenlieferung respektive die Auslieferung der Ergebnisdaten, wobei sich die „*Algorithmusentwicklung*“ auf die

Detailprobleme der Aufgabenstellung beschränkt. Der „Aufbau eines Massenprozessierungssystems“ konzentriert sich anschließend auf die „Massenproduktion“.

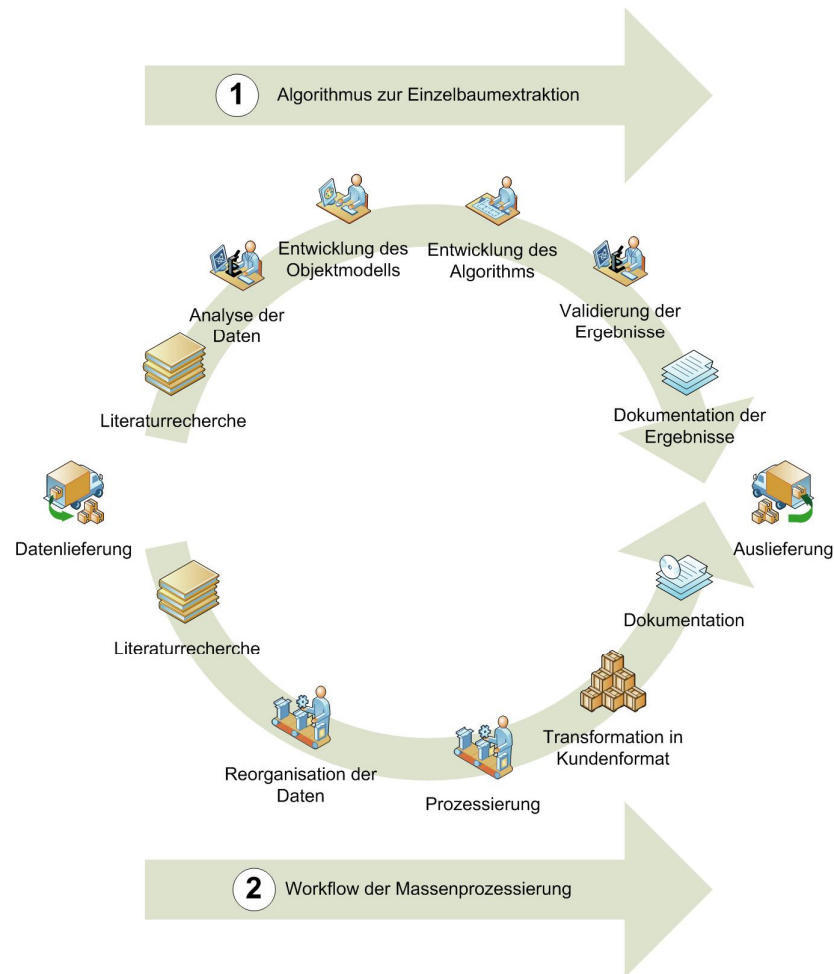


Abbildung 1: Methodische Vorgehensweise in den beiden Teilen der Arbeit

Das Ergebnis der vorliegenden Arbeit ist in den beiden voneinander getrennten Kapiteln 5 und 6 dokumentiert. **Kapitel 5** beschreibt den realisierten Algorithmus und die Bewertung des Algorithmus anhand einer Genauigkeitsanalyse. In **Kapitel 6** werden die Komponenten „Katalogisierung“ und „Verteiltes Rechnen“ des Massenprozessierungssystems vorgestellt. Auch dieses Kapitel schließt mit einer Bewertung, in der die Verarbeitungsgeschwindigkeit (Performance) des verteilten Systems derjenigen eines zentralen Multicore-Systems gegenübergestellt wird.

Abschließend enthält das **Kapitel 7** einer Diskussion der Validierungs-Ergebnisse und einem Ausblick.

2 Stand der Technik

2.1 Airborn Laserscanning

2.1.1 Messverfahren

Die flugzeuggetragene Laserscannermessung, auch Airborne Laser Scanning (ALS) genannt, basiert auf der Methode der optischen Abstandsmessung mittels Light detection and ranging (LIDAR). LIDAR ist ein aktives Fernerkundungsverfahren zur topographischen Erfassung der Geländeoberfläche. Das Verfahren besteht aus den Systemkomponenten Laserscanner und einer Kopplung von Differential-GPS (DGPS) und Inertial Navigation Systems (INS).

Die Höhenmessung der Geländeoberfläche erfolgt durch ausgesendete Laserimpulse, die quer zur Flugrichtung abgelenkt werden. Die Oberfläche wird dabei in einem definierten Geländestreifen abgetastet. Die Distanz zur Erdoberfläche wird über Laufzeitmessung bestimmt (Wever & Lindenberger 1999). Dabei wird die Zeitspanne gemessen, die zwischen der Aussendung eines Lichtpulses und der Registrierung des von einem Objekt reflektierten Lichtpulses verstreicht. Die präzise Lagebestimmung des Flugzeugs bzw. des Sensors und somit auch der gemessenen Laserpunkte wird durch das Zusammenwirken von GPS- und INS-Messungen erreicht.

In den Anfängen des ALS konnte mit den verfügbaren Sensoren lediglich die Laufzeit einer einzigen Pulsantwort pro ausgesendetem Signal detektiert werden (Wever & Lindenberger 1999). Inzwischen sind Messverfahren üblich, in denen zwei Pulsantworten (*first pulse, last pulse*) unterschieden werden können. Neuere Entwicklungen gehen über zur Erfassung und Auswertung der vollständigen Pulsantwort (*full waveform*). Krzystek und Reitberger (Reitberger et al. 2009) verwenden diese Daten bereits in Verfahren der automatisierten Baumextraktion. Abbildung 2 veranschaulicht den Unterschied zwischen einem diskreten Signal (Abbildung 2, links) und der vollständigen Pulsantwort (Abbildung 2, rechts).

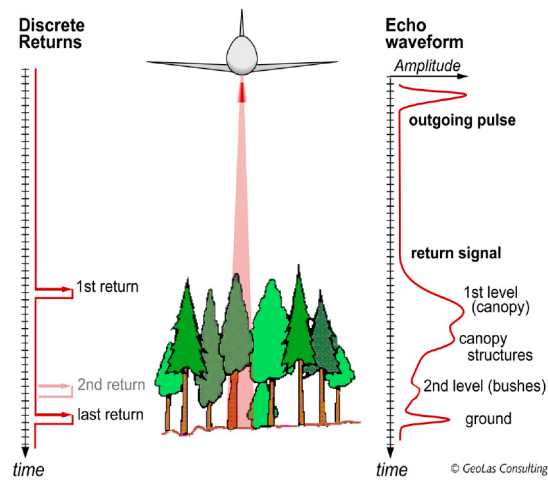


Abbildung 2: Gegenüberstellung von first pulse/last pulse und full waveform (Quelle: GeoLas Consulting)

Eine detaillierte Einführung, die alle Aspekte des flugzeuggetragenen Laserscan-Verfahrens vom Messprinzip über die Datenerfassung, mögliche Fehlerquellen der Datenerfassung, die Datenaufbereitung bis hin zur Verarbeitung umfasst, kann beispielsweise bei Hailer (Hailer 2005) oder Strellen (Strellen 2003) nachgelesen werden. Strellen verweist insbesondere auf die Eigenschaft von ALS-Daten als räumliche Punktwolken, deren „*Einzelmessungen sich mehr oder weniger zufällig über einen Flugstreifen verteilen*“ ohne expliziten Zusammenhang zwischen der Lage des gemessenen Einzelpunkts und des abgebildeten Objekts. Diese Besonderheit von ALS-Daten hat einen wesentlichen Einfluss auf die Verfahren, wie solche Daten verarbeitet werden können. Strellen (Strellen 2003) stellt dazu in seiner Arbeit ein Verfahren zur automatischen Berechnung des Geländemodells aus den Laser-Rohdaten vor. Dabei wird aus den Aufzeichnungen der First Pulse und der Last Pulse Reflexionen ein Digitales Oberflächenmodell (DOM) und ein Digitales Geländemodell (DGM) extrahiert. Abbildung 3 veranschaulicht die Trennung zwischen DOM (grün) und DGM (braun). Im Bereich zwischen den Bäumen sind DOM und DGM gleich.

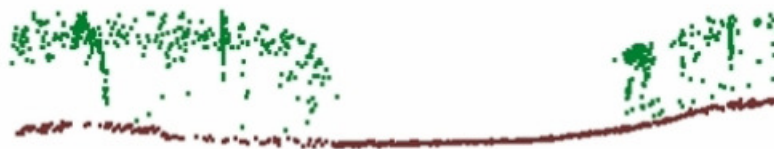


Abbildung 3: Schematische Darstellung des DOM (grün) und DGM (braun); (Quelle: Topscan)

Bei Schleyer (Schleyer 2001) ist das Verfahren beschrieben, das bei der Erstellung des DGM in Baden Württemberg angewendet wurde.

2.1.2 Reflexionsintensität

Der Laserimpuls wird in der Regel vom Objekt nicht gerichtet, sondern abhängig von der Beschaffenheit der Objektoberfläche diffus über den Raum verteilt zurückgestrahlt. Helle Oberflächen besitzen eine höhere Reflektivität als dunkle. Die Reflektivität natürlicher Oberflächen wird von Wever und Lindenberger (Wever & Lindenberger 1999) mit folgenden Werten angegeben:

- Sand 10% - 20%
- Vegetation: 30% - 50%
- Eis und Schnee: 50% - 80%

Die Reflektivität entscheidet somit über die quantitative Menge an Reflexionen, die vom Sensor wieder aufgefangen werden können, sowie über die Stärke des zurückgestrahlten Laserlichts. Moderne Laserscannersysteme sind in der Lage, die Signalstärke (Intensität) des von einem Messobjekt reflektierten Laserpulses zu vermessen und aufzuzeichnen (beispielsweise TopoSys 2009).

2.2 Bildanalyse zwischen Fernerkundung und Geoinformatik

2.2.1 Bildverstehen

Bildverstehen beschreibt den Prozess der rechnergestützten Lösung von Aufgabenstellungen, die sich an den Fähigkeiten des menschlichen visuellen Systems orientieren (Pinz 1994). Dabei werden zwei Strategien unterschieden. Bei der *Bottom-Up-Methode* wird ausgehend von den vorliegenden Bilddaten ein Bildmodell entwickelt. Mit Hilfe von Bildanalyseverfahren werden Objekte, die dem Bildmodell genügen, aus den Bilddaten extrahiert.

Bei der *Top-Down-Methode* wird dagegen aus dem Wissen über realweltliche Objekte ein Objektmodell entwickelt und dieses mit Hilfe von Regeln abgebildet auf deren Grundlage die Bilddaten analysiert werden. Abbildung 4 veranschaulicht die unterschiedliche Ausgangslage von Bottom-Up- und Top-Down-Methoden

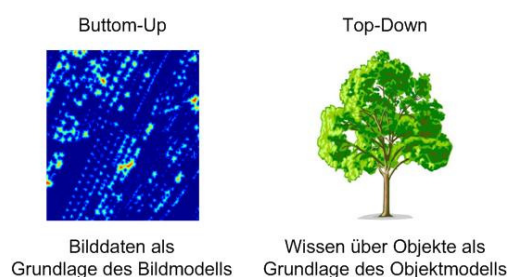


Abbildung 4: Unterschiedliche Ausgangslage von Bottom-Up- und Top-Down-Methoden

Viele Ansätze zur Extraktion von Objekten aus Bildern sind von der frühen Arbeit von Marr (Marr 1982) geprägt. Marr versucht in seiner Theorie die informationsverarbeitenden Schritte beim Sehen in einen Rahmen zu passen, der Physiologie, Psychologie und Künstliche Intelligenz umspannt. Dort wird der Vorgang der Objektextraktion gegliedert in eine „erste rohe Skizze“ (*Primal Sketch*), eine 2.5-D Skizze (*2.5-D Sketch*) und eine 3-D Modell Repräsentation (*3-D Model Representation*).

Die erste rohe Skizze enthält Bildmerkmale wie Kanten und Regionen, die direkt aus den Bildern abgeleitet werden können. Die 2.5-D Skizze enthält die aus den Bildern abgeleiteten räumlichen Oberflächen der Objekte, die ebenfalls noch als Objekt-Primitive (Punkte, Linien, Flächen) angesehen werden können. Im letzten Schritt werden durch Gruppierung der Primitive Instanzen der 3-D Objekte gebildet.

Eine alternative Betrachtung des Vorgangs der Objektextraktion unterscheidet zwischen modellgetriebenen und datengetriebenen Ansätzen. Bei den modellgetriebenen Ansätzen werden aus einem 3-D Objektmodell ein oder mehrere 2-D Muster, die im Bild gesucht werden. Das Objektmodell wird also ins Bild projiziert und daraus eine Objekthypothese aufgestellt und bewertet. Bei den datengetriebenen Ansätzen werden zunächst Merkmale wie Kanten und Regionen aus dem Bild extrahiert und erst in weiteren Verarbeitungsschritten zu höherwertigen Strukturen zusammengefasst, bis eine Zuordnung zum Modell möglich ist (Hinz 2004). Straub (Straub 2003) gibt eine ausführliche Übersicht der unterschiedlichen Theorieansätze der Objektextraktion.

2.2.2 Digitale Bildverarbeitung in der Fernerkundung

Die digitale Bildverarbeitung (*digital image processing*) ist im engeren Sinne ein Teilgebiet der Signalverarbeitung, im weiteren Sinne ein interdisziplinäres Feld in Gebieten der Physik, Informatik, Geowissenschaften und Ingenieurwissenschaften. Eine umfassende Einführung in die digitale Bildverarbeitung bieten beispielsweise Jähne (Jähne 2005) oder Gonzalez (Gonzalez & Woods 2006)).

Ursprünglich wurden in der Fernerkundung aus digitalen Bildern nur thematische Karten erstellt. Das Produkt war also die resultierende Karte. Im Gegensatz dazu stellen in der Geoinformatik die Daten selbst das Produkt dar. Thematische Darstellungen dieser Daten sind lediglich zusammenfassende Abbildungen einer Teilmenge der Daten.

Mit der Entwicklung immer besserer Sensoren und der damit einhergehenden Verbesserung der Auflösung von niedrig aufgelösten Rasterdaten (*low resolution*) hin

zu hoch aufgelösten Rasterdaten (*high resolution*) haben sich auch die Verfahren der Bilderverarbeitung weiterentwickelt. Abbildung 5 stellt die Entwicklung von niedrigen zu hochauflösenden Sensoren dem Objektdurchmesser gegenüber. Als Vergleich dienen die drei Objektarten Wald, Baumgruppe und Einzelbaum (Blaschke, Burnett & Pekkarinen 2004). In der Nähe der eingezeichneten Diagonalen entspricht der Objektdurchmesser der Größe einzelner Bildelemente (Pixeln). Im Bereich „*low resolution*“ sind die Bildelemente größer als der Objektdurchmesser. Einzelne Objekte können nicht sauber voneinander getrennt werden. Es kommt zum Phänomen der Mischpixel (Blaschke, Burnett & Pekkarinen 2004).

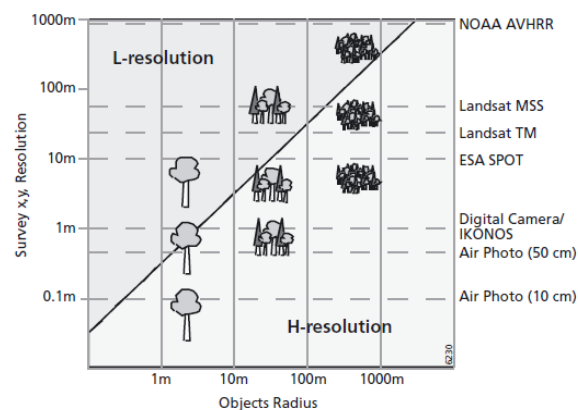


Abbildung 5: Beziehung zwischen Auflösung und der Größe realweltlicher Objekte

Im Bereich der hochaufgelösten Bilddaten können nicht nur einzelne Objekte und Objektarten besser voneinander getrennt werden. Aufgrund der hohen Auflösung können auch je nach Objektgröße einzelne Details oder Unterstrukturen unterschieden werden. Diese Informationsdichte erfordert jedoch die Entwicklung und Anwendung neuer Bildanalysemethoden mit denen es möglich ist, diese Unterstrukturen von Objekten zu identifizieren und zu trennen (Blaschke, Burnett & Pekkarinen 2004). Dabei sind zwei methodische Entwicklungen hervorzuheben.

Bei den Segmentierungsverfahren führte die Entwicklung der „*multiskalaren Segmentierung*“ zu neuen Methoden der Objektextraktion und Objektidentifikation. Die Verwendung unterschiedlicher Skalen bzw. Maßstäbe basiert auf der Überlegung, dass unterschiedliche Merkmale (Beispiel Straße) am besten in unterschiedlichen Maßstäben detektiert werden können. Unterstrukturen einer Straße, wie beispielsweise Fahrbahn- oder Randmarkierungen, werden idealerweise aus hochauflösenden Bildern extrahiert und liefern zusätzliche Merkmale des Objekts „Straße“. Dagegen ist im kleinen Maßstab - bei einer Auflösung, in der Straßen nur einige Pixel breit sind - der Netzcharakter von Straßen wesentlich deutlicher zu erkennen. Kleinere isolierte Objekte

wie einzelne Fahrzeuge oder Bäume beeinflussen hier die Extraktion nicht so stark wie in der hohen Auflösung.

Bei den Analyseverfahren hat sich die Kombination und Fusion verschiedener Datenquellen aus Fernerkundung und GIS etabliert. Auf diese Verfahren wird im folgenden Kapitel (Kapitel 2.2.3) näher eingegangen.

2.2.3 Wissen und wissensbasierte Bildanalyse

Wissensbasierten Bildanalysemethoden

Unter wissensbasierten Bildanalysemethoden werden Methoden verstanden, die explizites Wissen und Erfahrungen in den Prozess der Identifikation räumlicher Muster einbeziehen. Wissensbasierte Systeme integrieren dabei verschiedenste Formen von Wissen und finden in den unterschiedlichsten Disziplinen wie Biologie, Medizin, der Industrie aber auch in den Geowissenschaften Anwendung. Grundsätzlich kann zwischen dem existierenden Wissen in Form von Daten und dem Wissen des Bearbeiters bzw. Experten unterschieden werden.

In den Geowissenschaften wird vermehrt die Kombination existierenden Wissens in Form von Fernerkundungsdaten mit bereits bestehenden Daten in Form von Geoinformationen genutzt um neues Wissen und neue Informationen zu gewinnen (siehe Abbildung 6).

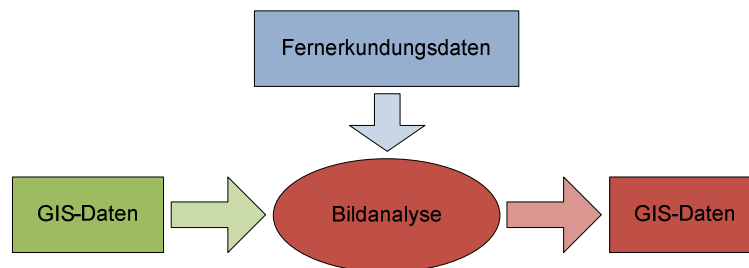


Abbildung 6: GIS-Daten als zusätzliches Ausgangs- und Ergebnisprodukt der fernerkundlichen Bildanalyse

Diese Vorgehensweise wird mit dem englischen Begriff „*data fusion*“ bezeichnet, einer „*integrierte Wissensverarbeitung*“ unterschiedlicher Informationsquellen. Die integrierte Auswertung von Fernerkundungs- und GIS-Daten ist bereits seit vielen Jahren intensiver Gegenstand der Forschung (de Kok 2001). Eine detaillierte Zusammenfassung dieser Entwicklung sowie eine darauf aufbauende Einführung in die Methode der objektorientierte Bildanalyse anhand eines konkreten Anwendungsbeispiels kann beispielsweise bei Mott (Mott 2005) nachgelesen werden.

Die vorliegende Arbeit nutzt ebenfalls diese Methode der integrierten Wissensverarbeitung. Mit Hilfe existierenden Wissens in Form von Liegenschaftsinformationen (GIS-Daten) wird ein Kontextmodell definitiv, innerhalb dessen das entwickelte Objektmodell „Streuobstbau“ angewendet werden kann. Im Umkehrschluss führt die Anwendung dieses Objektmodells außerhalb des Kontexts zu keinem sinnvollen Ergebnis.

Objektorientierte Bildanalyse in der Fernerkundung

Die Anwendung der Prinzipien wissensbasierter Systeme auf die Erfordernisse der digitalen Bildverarbeitung in der Fernerkundung führt zu der Entwicklung der „objektorientierten Bildanalyse“. Die objektorientierte Bildanalyse kann als eine Fortführung der automatisierten Auswertungsverfahren von Fernerkundungsdaten (z.B. überwachte und unüberwachte Klassifikation) betrachtet werden.

Merkmale wie „in der Nähe von“, „grenzt an“ oder „ist Teil von“ werden von einem Interpreten im Bild sofort erkannt. Klassische Bildanalyseverfahren können diese Informationen jedoch nicht nutzen, da diese räumlichen Abhängigkeiten nicht erfasst werden. Die objektorientierte Bildanalyse bietet jedoch Ansätze, auch räumliche Abhängigkeiten in Segmentierung und Klassifikation einfließen zu lassen (de Kok 2001).

2.3 Ausgewählte Themen der Bildverarbeitung

2.3.1 Delaunay-Triangulation

Die Delaunay-Triangulation wird in der vorliegenden Arbeit verwendet, um Nachbarschaftsbeziehungen innerhalb Laserscan-Punkte von Segmenten zur bestimmen.

Die Delaunay-Triangulation ist eine Methode zur Erstellung eines Dreiecknetzes aus einer Menge von Punkten. Dabei gilt für die zu bestimmenden Dreiecke die Bedingung, dass innerhalb eines Umkreises um drei Punkte, die ein Dreieck bilden, kein weiterer Punkt liegen darf. Diese Bedingung wird auch als „Kriterium des leeren Umkreises“ bezeichnet. Diese Methode liefert wohlgeformte Dreiecke (nahe an gleichseitigen Dreiecken). Die resultierenden Dreiecke überlappen sich nicht und können eine beliebig große Punktmenge flächendeckend und einheitlich vermaschen. Die Delaunay-Triangulation findet beispielsweise Anwendung bei Digitalen Geländemodellen und bildet dort die TIN-Datenstruktur (Triangular Irregular Network) im Gegensatz zur

Grid- bzw. Rasterdatenstruktur. Das Konstruktionsprinzip der in dieser Arbeit verwendeten Delaunay-Triangulation kann bei Barber nachgelesen werden (Barber, Dobkin & Huhdanpaa 1996)

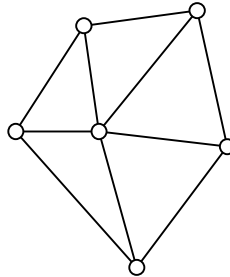


Abbildung 7: Delaunay-Triangulation

2.3.2 Räumliche Interpolationsverfahren

Ziel einer räumlichen Interpolation ist die Generierung einer Oberfläche oder eines Oberflächenmodells aus diskreten Stützpunkten (Höhenmodell, Schadstoffverteilung). Es wird zwischen exakten und approximativen Interpolationsverfahren unterschieden. Bei den exakten Verfahren verläuft die resultierende Oberfläche exakt durch die zugrundeliegenden Stützpunkte und gibt an diesen Stellen die jeweiligen Messwerte wieder. Bei den approximativen Verfahren wird aus den Messpunkten eine Näherungsfläche berechnet. Somit ist es möglich, dass Fehler in den Ausgangsdaten geglättet werden können.

Ein sehr einfaches Interpolationsverfahren ist die „*Nearest-Neighbor-Interpolation*“. Dieses Verfahren wird häufig bei der Rasterung von Messwerten verwendet und dort im Speziellen zur Vervollständigung nicht belegter Rasterzellen auf Basis der Nachbarzellenwerte. Diese Methode findet auch Anwendung in der vorliegenden Arbeit.

Andere Interpolationsverfahren sind die „*Inverse Distanzgewichtung*“, bei der Fehlstellen durch das, umgekehrt proportionalen zum Abstand, gewichtete Mittel der benachbarten Messwerte geschätzt wird. Weitere verbreitete Verfahren sind die „*bilineare Interpolation*“, die „*Polynomregression*“, die „*Spline-Interpolation*“ oder die Familie der „*Kriging-Interpolationsverfahren*“, die auf statistischen Annahmen über den räumlichen Zusammenhang der Messpunkte basieren (Longley 2007).

2.3.3 Morphologische Mathematik

Die morphologische Mathematik ist eine Teildisziplin der Bildverarbeitung, die sich mit der Verarbeitung von binären Rasterdaten (Binärbildern) befasst. Ihre Ursprünge liegen

in der Mengenlehre. In einem Binärbild werden Objekte bzw. zusammenhängende Regionen mit einem Pixelwert = „1“ als Menge bezeichnet. Bei der morphologischen Mathematik handelt es sich nun um eine Klasse von Nachbarschaftsoperationen, die auf diese Mengen angewendet werden. So können beispielsweise kleine Lücken im Bild geschlossen oder kleine Störungen aus dem Bild eliminiert werden.

Die Funktionsweise der morphologischen Mathematik basiert auf der binären Faltung. Dabei wird eine $(2R+1) \times (2R+1)$ -Maske pixelweise über das gesamte Binärbild verschoben. Befinden sich ein oder mehrere Objektpixel (Pixelwert = „1“) innerhalb der Maske, ist das Ergebnis der Operation eins, ansonsten null. Das Objekt wird also ausgedehnt. Der so beschriebene Operator wird Dilatationsoperator genannt. Die Größe der Maske hat dabei wesentlichen Einfluss auf die Wirkung des Operators. Daher wird die Maske häufig auch als Strukturelement (SE) bezeichnet.

Bei einem Binärbild gibt es bezüglich der Art durchzuführender Operationen nur wenige Möglichkeiten. Binäre Bildpunkte lassen sich nur mit den logischen Operatoren der Booleschen Algebra kombinieren. Damit lassen sich alle Operationen der morphologischen Mathematik auf die beiden Basisoperationen „Und“ (entspricht der Erosion; siehe im Folgenden) und „Oder“ (entspricht der Dilatation) reduzieren. Aus diesen Basisoperationen lassen sich jedoch vielseitige zusammengesetzte Operationen ableiten (Soille 2003, Jähne 2005).

Morphologische Basisoperationen

Die Dilatation eines Objektes $f(\vec{x})$ durch das Strukturelement $g(\vec{z})$ wird definiert durch die Funktion:

$$(f \oplus g) = \max_{\vec{x} \in G} \{f(\vec{x} - \vec{z}) + g(\vec{z})\}$$

Formel 1: Dilatation

Die Erosion \ominus eines Objektes $f(\vec{x})$ durch das Strukturelement $g(\vec{z})$ wird definiert durch die Funktion:

$$(f \ominus g) = \min_{\vec{x} \in G} \{f(\vec{x} + \vec{z}) - g(\vec{z})\}$$

Formel 2: Erosion

Dilatation und Erosion sind in Abbildung 8 schematisch dargestellt.

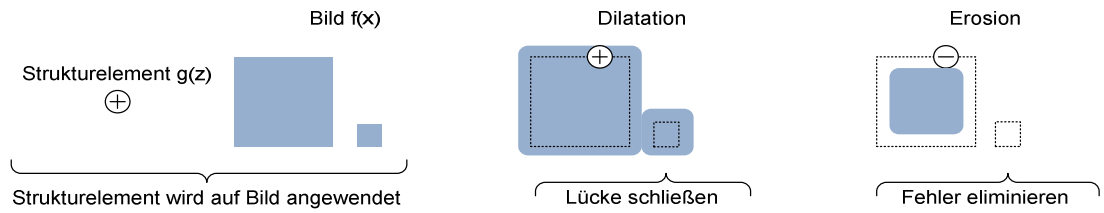


Abbildung 8: Morphologische Basisoperationen Dilatation und Erosion

Zusammengesetzte morphologische Operationen

Unter Verwendung der elementaren Basisoperationen Erosion und Dilatation können weitere Operationen für die Bearbeitung von Objektformen entwickelt werden. Die Erosion wird zur Entfernung kleiner Objekte verwendet. Sie hat allerdings den Nachteil, dass alle im Bild verbleibenden Objekte kleiner werden. Durch eine anschließende Dilatation des Bildes mit dem gleichen Strukturelement lässt sich dies vermeiden. Die Kombination aus Erosion und Dilatation wird Öffnen (Opening) genannt. Das Öffnen eliminiert alle Objekte, die das Strukturelement in keinem Punkt vollständig enthalten und vermeidet gleichzeitig die Größenreduktion aller verbleibenden Objekte:

$$(f \circ g) = (f \ominus g) \oplus g$$

Formel 3: Opening

Die Dilatation vergrößert Objekte und füllt kleine Löcher und Risse. Die damit einhergehende, generelle Vergrößerung aller Objekte kann durch eine anschließende Erosion ausgeglichen werden. Die Kombination von Dilatation und Erosion wird Schließen (Closing) genannt:

$$(f \bullet g) = (f \oplus g) \ominus g$$

Formel 4: Closing

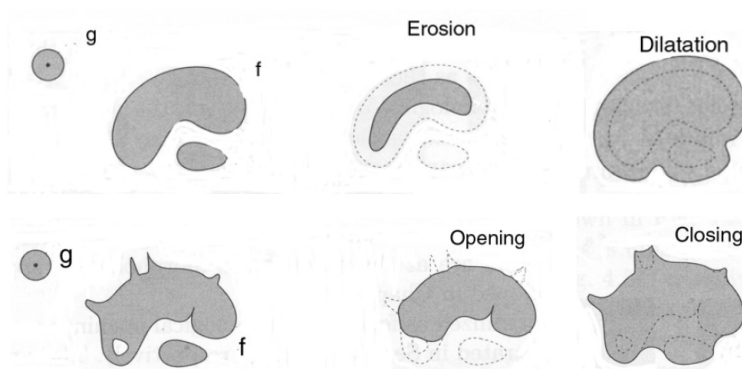


Abbildung 9: Gegenüberstellung der Basisoperationen und der zusammengesetzten Operationen (Soille 2003)

Weitere Zusammengesetzte morphologische Operationen sowie verschiedene Algorithmen, die auf diesen Operationen basieren sind bei Gonzalez (Gonzalez & Woods 2006) detailliert beschrieben.

2.3.4 Segmentierung

Die Segmentierung ist ein Teilgebiet der digitalen Bildverarbeitung und des „*maschinellen Sehens*“. Sie ist die Basis für nachfolgende Verarbeitungsschritte wie Objektextraktion oder Klassifikation. Dabei ist die Segmentierung von großer Bedeutung, da die Qualität der durch die Segmentierung generierten Objekte direkt die Qualität der Folgeschritte beeinflusst.

Es existieren verschiedene Definitionen, die den Prozess der Segmentierung unterschiedlich präzise beschreiben. Blaschke fasst die Definition wie folgt zusammen: „*Die rechnergestützte Segmentierung unterteilt das digitale Bild in Bereiche (Segmente, räumlich zusammenhängende Mengen von Pixeln), die in der realen Welt eine Bedeutung haben. Die Segmentierung soll je nach Untersuchungszweck relevante Bildelemente bzw. repräsentierte Landschaftselemente ergeben.*“ (Blaschke 2000).

Segmentierungsverfahren finden heute in vielen Bereichen wie Medizin, Neuroinformatik, Nachrichtentechnik und Fernerkundung Anwendung. Oczipka gibt in seiner Arbeit eine Übersicht, nicht nur über unterschiedliche Segmentierungsverfahren sondern auch über deren Einteilung (Oczipka 2007). Eine verbreitete Einteilung unterscheidet zwischen pixelorientierten, kantenbasierten und regionenbasierten Verfahren (Jähne 2005, Blaschke, Burnett & Pekkarinen 2004).

Pixelorientierte Segmentierungsverfahren

Pixelorientierte Verfahren betrachten nur den einzelnen Grauwert eines Pixels ohne Berücksichtigung von Nachbarschaftsbeziehungen. Das einfachste pixelbasierte Verfahren ist der globale Schwellwert (*threshold*). Dabei werden alle Pixel mit Grauwerten über einem zuvor bestimmten Schwellwert t dem Vordergrund, alle anderen dem Hintergrund zugewiesen (Jähne 2005). Der Schwellwert wird dabei in der Regel aus dem Histogramm ermittelt.

Kantenbasierte Segmentierungsverfahren

Kantenbasierte Verfahren nutzen zur Segmentbildung Diskontinuitäten im Bild, also starke Änderungen des Grauwertes oder einen Höhengsprung im Oberflächenmodell.

Diese Kanten stellen die Segmentgrenzen dar und werden in weiteren Verarbeitungsschritten (Konturverfolgung) zu Objekten verbunden. Die meisten kantenbasierten Verfahren setzen eine dreistufige Vorverarbeitung voraus: Filterung, Verstärkung und Erkennung (Blaschke, Burnett & Pekkarinen 2004).

Regionenbasierte Segmentierungsverfahren

Regionenbasierte Verfahren erweitern die vorherigen Verfahren um den entscheidenden Aspekt, dass Objekte stets zusammenhängend und somit voneinander abgrenzbar sind. Neben dem Grauwert ist also auch die relative Anordnung der Pixel zueinander von Bedeutung (Nachbarschaft). Diese Verfahren versuchen möglichst homogene Regionen im Bild zu finden. Gonzalez stellt dazu ein Regelwerk von Bedingungen auf, die jeder Algorithmus zur regionenbasierten Segmentierung erfüllen muss (Gonzalez & Woods 2006).

Die gängigsten regionenbasierten Verfahren sind Region-Splitting, Region-Merging und Region-Growing. Beim Region-Splitting wird das gesamte Bild als Initialsegment verwendet und dieses iterativ in immer kleinere Segmente zerlegt, solange das Homogenitätskriterium (Abbruchkriterium) der Segmentierung noch erfüllt ist (Gonzalez & Woods 2006). Beim Region-Merging wird das Ausgangsbild zunächst übersegmentiert. Anschließend werden benachbarte Regionen solange verschmolzen, bis auch hier wiederum das Abbruchkriterium erfüllt ist (Gonzalez & Woods 2006). Region-Merging ist auch Grundlage des Segmentierungsverfahrens der Definiens-Software. Region-Growing-Verfahren werden auch als Flächenwachstumsalgorithmus bezeichnet. Zunächst werden so genannte Saatpunkte (seed pixel) als Anfangssegmente definiert. Wie diese Saatpunkte gefunden werden, ist unabhängig vom Algorithmus und hängt von der jeweiligen Datengrundlage und dem Objektmodell ab. Ausgehend von den Saatpunkten werden diese mit den jeweils benachbarten Punkten verschmolzen, sofern diese noch keinem anderen Segmente zugeordnet sind. Die Segmente werden dabei auf Ähnlichkeit bzw. Homogenität geprüft. Auch diese Prüfung ist wiederum abhängig von Datengrundlage und Objektmodell. Die Anzahl der Segmente und deren Größe sind abhängig von dem anwendungsfallspezifischen Verfahren, mit dem sie Saatpunkte ausgewählt werden und

Multiskalare Segmentierung

Die multiskalare Segmentierung ist ein spezielles Region-Merging-Verfahren, das mit einzelnen Pixeln als Startobjekten beginnt. In nachfolgenden Schritten wird jeweils ein

Paar von Bildobjekten zu immer größeren Objekten zusammengefasst. In jeder Verschmelzung wird jedes Segment nur einmal berücksichtigt. Diese Vorgehensweise wird angewandt, um eine möglichst ähnliche Größe der Segmente sicherzustellen. Das Verfahren wurde von der Firma Definiens Imaging entwickelt und hat sich inzwischen in vielen Bereichen durchgesetzt (Baatz & Schäpe 2000). Die multiskalare Segmentierung erweist sich im Besonderen dort als nützlich, wo Objektmerkmale auf unterschiedlichen Skalenniveaus segmentiert und extrahiert werden. So können Straßen in einer groben Auflösung als einfache Linie detektiert werden. Bei einer feineren Auflösung können schließlich einzelnen Fahrbahnstreifen unterschieden werden. Auf der groben Ebene wird also nur das Objekt Straße erkannt. In der feinen Darstellung kann zwischen unterschiedlichen Straßentypen (Landstraße, Autobahn) unterschieden werden.

Morphologische Segmentierung: Watershed-Transformation (WST)

Bei der Watershed-Transformation, die auch bei der Entwicklung des Algorithmus dieser Arbeit Verwendung findet, wird das Bild zunächst als Grauwertgebirge betrachtet (Vincent & Soille 1991). Das Verfahren kann wie folgt erklärt werden: Fällt ein Wassertropfen auf das Grauwertgebirge, so fließt er nach unten, bis er ein lokales Minimum erreicht hat. Das Einzugsgebiet eines Minimums wird als Auffangbecken (catchment basins) bezeichnet, die Grenzen zwischen den Auffangbecken als Wasserscheiden (watersheds). Bei der Watershed-Transformation werden die Pixel um ein regionales Minimum zu einem Segment zusammengefasst. Häufig muss das Ausgangsbild zunächst transformiert werden, damit der WST-Algorithmus darauf angewendet werden kann (Soille 2003). Im vorliegenden Fall der Einzelbaumextraktion muss das Ausgangsbild (Oberflächenmodell) zunächst invertiert werden, damit die Baumwipfel als Täler interpretiert werden und somit vom WST-Algorithmus korrekt segmentiert werden.

2.3.5 Rasterdatenzerlegung

Ein Parallelisieren der Verarbeitung umfangreicher Datenmengen wie in der vorliegenden Arbeit setzt voraus, dass das Problem in unabhängige Teilprobleme zerlegt werden kann. Dabei hat der Zerlegungsprozess wesentlichen Einfluss sowohl auf die Gesamtlaufzeit als auch auf das Ergebnis der Verarbeitung.

Reguläre Gitter

Reguläre Zerlegung (*regular decomposition*) ist nach Zhang (Zhang et al. 2005) die Aufteilung großräumiger Rasterdaten in Blöcke gleicher Spalten- und Zeilenzahl. Zentrales Problem bei der Zerlegung solcher Blöcke ist die Behandlung der Randbereiche in denen sich Verarbeitungsergebnisse wie beispielsweise Baumkonturen überschneiden (siehe Abbildung 10).

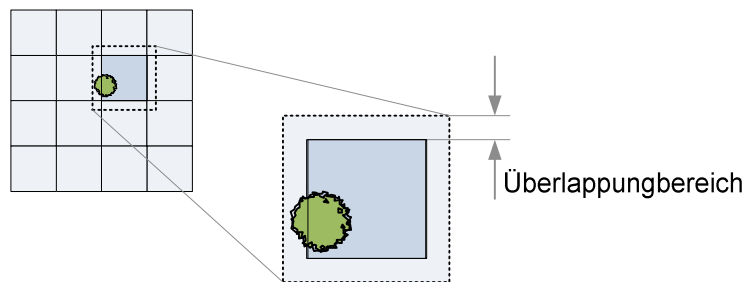


Abbildung 10: Objekt außerhalb des Blocks aber innerhalb des Überlappungsbereichs

Es stellt sich die Frage wie das Verarbeitungsergebnis einem bestimmten Block eindeutig zugeordnet werden kann. Abbildung 11 veranschaulicht dieses Problem nochmals für den Fall der Objektextraktion. Auf der linken Seite werden aufgrund der Zerschneidung in Blöcke zwei getrennte Objekte identifiziert. Auf der rechten Seite kann das Objekt als ganzes identifiziert und mittels des Zentroids eindeutig einem Block zugeordnet werden. Im konkreten Fall der Einzelbaumextraktion erfolgt die eindeutige Blockzuordnung durch die punktförmige Baumstammposition.

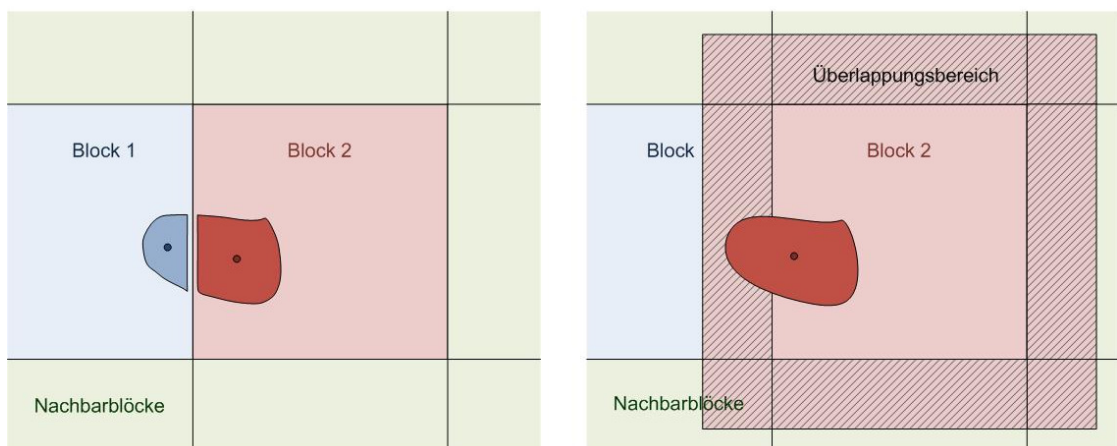


Abbildung 11: Rasterdatenzerlegung ohne (links) und mit Überlappungsbereich (rechts)

Die Größe des Überlappungsbereichs ist von der Größe der zu extrahierenden Objekte abhängig. Die Wahl der Blockgröße ist abhängig von der Komplexität des Algorithmus und damit vom Ressourcenbedarf.

Werden innerhalb des Algorithmus globale Filter oder Operatoren verwendet, also Operatoren, die alle Daten des Blocks berücksichtigen (beispielsweise die Delaunay-Triangulation), so muss bei der Wahl der Blockgröße auf die Zeitkomplexität der Operation geachtet werden. Der in der vorliegenden Arbeit verwendete Matlab-Algorithmus basiert auf einer von Barber et al. (Barber, Dobkin & Huhdanpaa 1996) entwickelten Variante der Delaunay-Triangulation mit der Komplexität $O(n \times \ln(n))$ (vgl. Kapitel 2.3.1). Der Rechenaufwand steigt demnach mit zunehmender Blockgröße überproportional an.

2.3.6 Bewertungsverfahren für Segmentierungs- und Klassifikationsergebnisse

Qualitätsmaße überwachter und unüberwachter Klassifikationsverfahren

Die Methode des **Accuracy Assessment** stellt ein Standardwerkzeug zur Beurteilung der Qualität beispielsweise von überwachten und unüberwachten Klassifikationen in der Fernerkundung dar. Zentrales Element dieser Methode ist die Konfusions- oder Fehlermatrix. Diese gibt in Form einer tabellarischen Auflistung Aufschluss über die Qualität einer Klassifizierung. Aus der Konfusionsmatrix kann das Maß der **Gesamtgenauigkeit** (*overall accuracy*, O_A) abgeleitet werden. Die Gesamtgenauigkeit gibt die Prozentzahl der korrekt klassifizierten Pixel an und berechnet sich aus der Summe der Hauptdiagonalen (korrekt klassifizierte Pixel) dividiert durch die Gesamtanzahl der Referenzen. Die Gesamtgenauigkeit liefert jedoch noch keine Informationen über die Qualität der einzelnen Klassen. Für eine detailliertere Qualitätsbeschreibung der einzelnen Objektklassen können aus der Matrix die **Nutzergenauigkeit** (*user's accuracy*, U_A) und **Herstellergenauigkeit** (*producer's accuracy*, P_A) je Klasse berechnet werden (Congalton & Green 1999).

Ein weiteres Maß, das aus der Konfusionsmatrix abgeleitet werden kann ist der **Kappa Index**. Bei einer vollständigen Übereinstimmung zwischen Klassifizierung und Referenzdaten ist der Wert des Kappa Index gleich „1“, bei gleichen Anteilen von korrekten und inkorrekten Übereinstimmungen „0“ und bei keinerlei Übereinstimmung „-1“. Der Kappa Index kann auch für einzelne Klassen der Konfusionsmatrix berechnet werden.

		Referenzdaten- klassen				Zeilen- summen
Klassen aus der Klassifizierung		n_{11}	n_{12}	n_{13}	n_{1k}	Σn_{1x}
		n_{21}	n_{22}	n_{23}	n_{2k}	Σn_{2x}
		n_{31}	n_{32}	n_{33}	n_{3k}	Σn_{3x}
		n_{k1}	n_{k2}	n_{k3}	n_{kk}	Σn_{4x}
Spalten- summen		Σn_{x1}	Σn_{x2}	Σn_{x3}	Σn_{x4}	

Abbildung 12: Konfusionsmatrix

$$\text{Overall Accuracy} = \frac{\sum_{i=1}^k n_{ii}}{n} \cdot 100$$

n_{ii} = Elemente der Hauptdiagonalen

n = Anzahl Referenzpixel

$$\text{User's Accuracy} = \frac{n_{ii}}{n_{ix}}$$

n_{ii} = korrekt klassifizierte Pixel der Klasse i

n_{ix} = Zeilensumme der Klasse i

$$\text{Producer's Accuracy} = \frac{n_{jj}}{n_{xj}}$$

n_{jj} = korrekt klassifizierte Pixel der Referenz j

n_{xj} = Spaltensumme der Referenz j

$$\text{Kappa Index} = \frac{n \sum_{i=1}^k n_{ii} - \sum_{i=1}^k n_{ix} \cdot n_{xj}}{n^2 - \sum_{i=1}^k n_{ix} \cdot n_{xj}}$$

k = Anzahl der Klassen

n = Anzahl der Referenzpixel

Formel 5: Gesamtgenauigkeit

Die Konfusionsmatrix hat sich weitgehend als Standard zur Beschreibung der Qualität von Klassifizierungen durchgesetzt, da sie die Übereinstimmung der Klassifizierung mit Referenzdaten oder Ground Truth belegt. Es gibt jedoch verschiedene Faktoren, die dieses Qualitätsmaße beeinflussen und als gegeben vorausgesetzt werden (Foody & Atkinson 2002).

- Die Referenzdaten werden als 100% fehlerfrei angenommen, was sicherlich auf keine real aufgezeichneten Referenzdaten zutrifft.
- Die Erfassung von Referenz- und Klassifizierungsdaten kann zu unterschiedlichen Zeiten erfolgt sein, was zu nicht quantifizierbaren Unterschieden führen kann.
- Eine meist nicht zutreffende Annahme ist das Vorliegen reiner Pixel. In der Realität gibt es jedoch neben Pixeln, die eindeutig einer Klasse zugeordnet werden können, auch Mischpixel.
- Die Konfusionsmatrix und die aus ihr abgeleiteten Fehlermaße begeben keine Auskunft über die räumliche Verteilung der Klassifizierungsfehler.

Qualitätsmaße automatischer Objektextraktionsverfahren

Zur Quantifizierung der Qualität der Extraktionsergebnisse werden die Maße **Vollständigkeit** und **Korrektheit** verwendet. Diese Maße ergeben sich wie folgt (siehe Abbildung 13). Zunächst wird die Anzahl der Objekte bestimmt, die richtig extrahiert wurden, also derjenigen Objekte, die sowohl in den Referenzdaten als auch in den

Extraktionsdaten enthalten sind. Diese Anzahl wird als True Positives (TP) bezeichnet. Das Extraktionsergebnis enthält neben den richtigen Ergebnissen aber auch fehlerhaft extrahierte Objekte, beispielsweise Gebäude oder Hochspannungsmasten, die fälschlicherweise als Bäume erkannt wurden. Die Anzahl dieser Objekte wird als False Positives (FP) bezeichnet. Die Menge der nicht extrahierten Objekte, also derjenigen Objekte, die in der Referenz enthalten sind, aber nicht extrahiert wurden, werden als False Negatives (FN) bezeichnet.

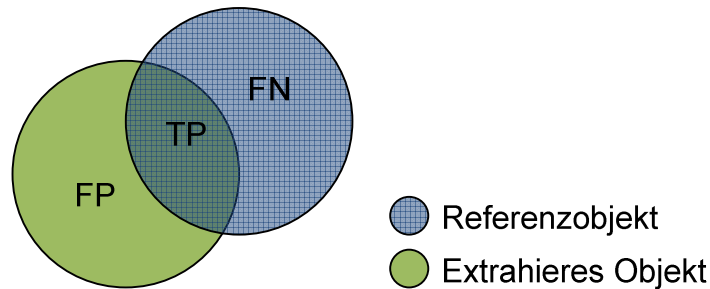


Abbildung 13: Anzahl der Objekte in Referenz und Extraktionsergebnis

Die Begriffe **Vollständigkeit** und **Korrektheit** sind im Bereich der Bewertung automatischer Objektextraktionsverfahren sehr verbreitet (vgl. Straub 2003). Im Bereich der Fernerkundung wird anstelle von Vollständigkeit von Herstellergenauigkeit (Producer's Accuracy) und anstelle von Korrektheit zumeist von Anwendergenauigkeit (User's Accuracy) gesprochen. Entsprechend wird häufig der Begriff Ommision-Error synonym zu False Negatives und Commision-Error für False Positives verwendet.

Die Referenzdaten enthalten somit TP + FN Objekte, das Extraktionsergebnis TP + FP Objekte. Auf der Basis dieser Werte ergibt sich die Vollständigkeit V als:

$$V = \frac{TP}{TP + FN}, 0 \leq V \leq 1$$

Formel 6: Qualitätsmaß Vollständigkeit

Die Korrektheit K des Extraktionsergebnisses errechnet sich aus dem Verhältnis der Anzahl korrekt extrahierter Objekte zur Anzahl aller extrahierten Objekte:

$$K = \frac{TP}{TP + FP}, 0 \leq K \leq 1$$

Formel 7: Qualitätsmaß Korrektheit

In dieser Arbeit werden die Begriffe Vollständigkeit und Korrektheit zur Charakterisierung der Qualität der Ergebnisse des Algorithmus zur Einzelbaumextraktion verwendet. Damit kann eine grobe Abschätzung der

Leistungsfähigkeit des Verfahrens erfolgen. Bei diesen Maßen ist jedoch zu beachten, wie die Zuordnung zwischen den extrahierten Objekten und den als Referenz angesehenen Objekten realisiert wird. Diese Zuordnungsvorschrift hat wesentlichen Einfluss auf die Parameter TP, FP und FN. Die Werte Vollständigkeit und Korrektheit dürfen demnach nur eine grobe Einschätzung des Ergebnisses darstellen und bedürfen in jedem Fall einer Diskussion.

2.4 Objektextraktion von Bäumen aus Fernerkundungsdaten

2.4.1 Der Begriff „Streuobst“

Streuobst hat eine lange Tradition, aber der Begriff „Streuobst“ ist relativ jung. Geschützt war und ist dieser Begriff nicht. Erst auf der Fachtagung „*Streuobst in der Kulturlandschaft 2006*“ der Bayerischen Landesanstalt für Landwirtschaft (LfL) hat man sich zunächst auf eine vorläufige Definition geeinigt (Girstenbreu 2006). Aus dieser Definition lassen sich weitere, anwendungsspezifische Annahmen ableiten, die wieder Einfluss auf das zu entwickelnde Objektmodell haben werden:

„Unter Streuobst versteht man Obst (Äpfel, Birnen, Süß- und Sauerkirschen, Beerenobst, Pflaumen/Zwetschgen, Pfirsiche und Aprikosen sowie einige Wildarten) von großwüchsigen Bäumen (Stammhöhe mindestens 1,60 m) verschiedener Obstarten, Sorten und Altersstufen, die in Gärten, an Ortsrändern, auf Feldern, Wiesen und Weiden in ziemlich unregelmäßigen Abständen gewissermaßen gestreut stehen. Auf den Einsatz chemischer Pflanzenschutz- oder Düngemittel wird in der Regel verzichtet. Streuobstbestände dürfen eine maximale Bestandsdichte von 100 Bäumen je ha haben. Sie sind eine extensiv genutzte Kombination von Hochstamm-Obstbäumen mit regelmäßiger Unternutzung (z.B. Wiese, Weide, oder Acker). Charakteristisch ist die Mischung verschiedener standortangepasster Obstarten und Obstsorten.“ (Girstenbreu 2006)

Ein weiterer Indikator, der allerdings noch nicht Teil der vorläufigen Definition ist, besagt, dass im Streuobstbestand Halbstämme mit einer Mindesthöhe von 1,20 m zu einem gewissen Prozentsatz erlaubt sind.

2.4.2 Bisherige Arbeiten zur Baumextraktion

Die automatische Extraktion von Einzelbäumen und Baumgruppen war bereits Inhalt zahlreicher Arbeiten. In frühen Arbeiten zu diesem Thema wurde ausschließlich auf

Farb- und Falschfarben-Luftbildern gearbeitet (Pinz 1989, Brandtberger & Walter 1998, Larsen & Rudemo 1998). Erst in den neunziger Jahren des letzten Jahrhunderts wurden digitale optische Sensoren (multispektral mapper, Gougeon 1995) verwendet. Mit der Kopplung von Differential-GPS (DGPS) und Inertial Navigation Systems (INS) wurde dann Ende der neunziger Jahre die Voraussetzung geschaffen, auch Laserscanner als Datenquelle für die Baumextraktion einsetzen zu können (Campbell 2007, Næsset et al. 2004).

Wie Bechtel (Bechtel 2007) in seiner Arbeit zusammenfasst, stammen viele Impulse der Objektextraktion, insbesondere für den Forstbereich, aus den skandinavischen Ländern. Dabei sind in den verschiedenen Ländern unterschiedliche Strategien und Schwerpunkte auszumachen. Während man sich in Finnland und Schweden auf Einzelbaumverfahren konzentriert hat (Hyypä & Inkinen 1999, Heurich et al. 2004), wurden in Norwegen vorwiegend Verfahren zur Bestimmung von Parametern von Baumbeständen aus ALS-Daten entwickelt. Eine detaillierte Übersicht über alle skandinavischen Arbeiten bietet Næsset et al. (Næsset et al. 2004).

Im deutschsprachigen Raum haben sich in den vergangenen Jahren verschiedene Forschungseinrichtungen ebenfalls intensiv mit der Baumextraktion beschäftigt. So wurden beispielsweise an der Universität Freiburg (Koch et al. 2009, Hyypä et al. 1999) umfangreiche Studien zur Baumextraktion im Rahmen des europäischen Projekts HIGHSCAN (*Assessing Forest Stand Attributes by Integrated Use of High-Resolution Satellite Data and Laserscanner*) durchgeführt. Desweiteren ist die TU München (Heurich et al. 2004, Heurich, Kennel & Kennel 2007) zu nennen.

In jüngster Zeit geht die Entwicklung hin zu vergleichenden Untersuchungen, wie bei Chang et al. (Chang et al. 2008) oder bei Bechtel (BECHTEL 2007), in denen unterschiedliche Segmentierungsverfahren gegenübergestellt und bewertet werden. Bechtel entwickelt ein möglichst vollständiges und umfassendes Baummodell, basierend auf den Modellen verschiedener Baumextraktionsverfahren, welches teilweise auch Grundlage der vorliegenden Arbeit ist.

Neueste Entwicklungen im Bereich der Laserscantechnik erlauben Systeme, die das gesamte reflektierte Signal detektieren (full waveform LIDAR). Es ist inzwischen möglich, Feinstrukturen innerhalb von Bäumen zu detektieren und somit detailliertere Merkmale zu extrahieren (Koch et al. 2009, Reitberger, Krzystek & Stilla 2008). Reitberger verwendet hierbei einen neu entwickelten 3-D-Segmentierungsalgorithmus.

In Abbildung 14 ist erkennbar, dass mit diesem Verfahren auch Unterstrukturen wie Unterholz klar von Hochstämmen abgegrenzt werden können.

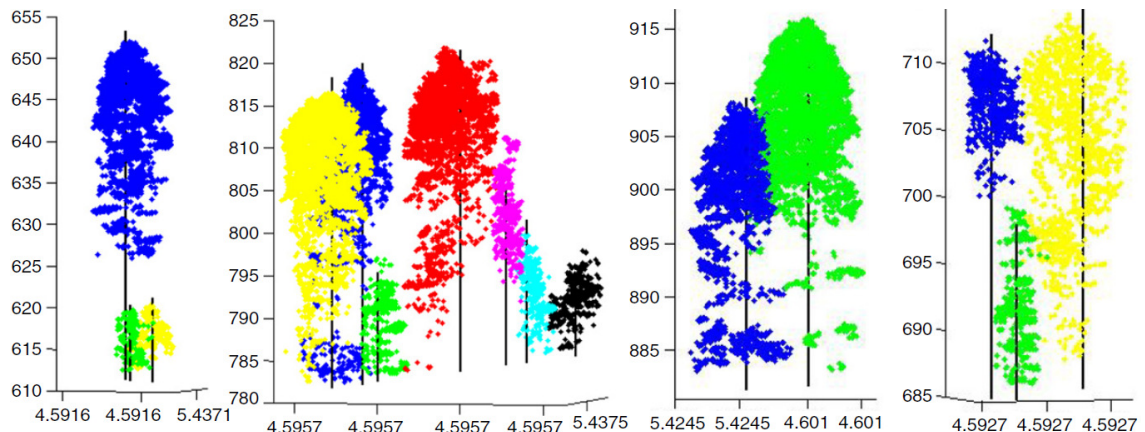


Abbildung 14: 3-D-Segmentierung auf Full Waveform LIDAR Daten (Reitberger et al. 2009)

Andererseits lässt sich ein weiterer Trend beobachten, in dem konventionelle ALS-Daten mit spektralen Daten kombiniert werden, um detaillierte und kostengünstige Produkte zu generieren oder einmal erfasste Basisdaten zu aktualisieren (Tiede, Osberger & Novak 2009).

Bei Bechtel (Bechtel 2007) findet sich eine Zusammenfassung und Gegenüberstellung aktueller Verfahren der Baumextraktion. Außerdem bewertet Bechtel diese Verfahren im Hinblick auf ihre Anwendbarkeit im Rahmen eines Expertensystems. Dabei werden für jedes Verfahren zunächst die wichtigsten Eigenschaften zusammengefasst. Grundsätzlich wird zwischen einzelbaumbasierten Verfahren (*individual tree crown, ITC*) und bestandsbasierten Verfahren (*stand-level*) unterschieden. Die Verfahren basieren jeweils auf unterschiedlichen Ausgangsdaten. Es kann zwischen Verfahren unterschieden werden, die auf der Basis von Höhenmodellen arbeiten und solchen, denen radiometrische Bilddaten zugrunde liegen. Ein weiteres Unterscheidungskriterium der Verfahren besteht in den unterschiedlichen Extraktionsparametern wie Baumhöhe, Kronendurchmesser oder Kronenbegrenzung.

Viele der Verfahren basieren auf dem Baummodell von Pollock (Pollock 1996). Dieses Modell beschreibt ein Ellipsoid, das mit drei Formparametern angenähert wird.

$$h(\vec{x}) = \left| \left(a^n \left(1 - \frac{(x^2 + y^2)^{\frac{n}{2}}}{b^n} \right) \right)^{\frac{1}{n}} \right|$$

$$\text{mit } \vec{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

a = Baumhöhe

b = Kronendurchmesser

n = Form des Ellipsoids

Formel 8: Baummodell nach Pollock

Der Parameter a entspricht der Baumhöhe, b dem Kronendurchmesser und n beschreibt die Form des Ellipsoids. Üblicherweise wird nur die obere Hälfte des Ellipsoids als Modell verwendet.

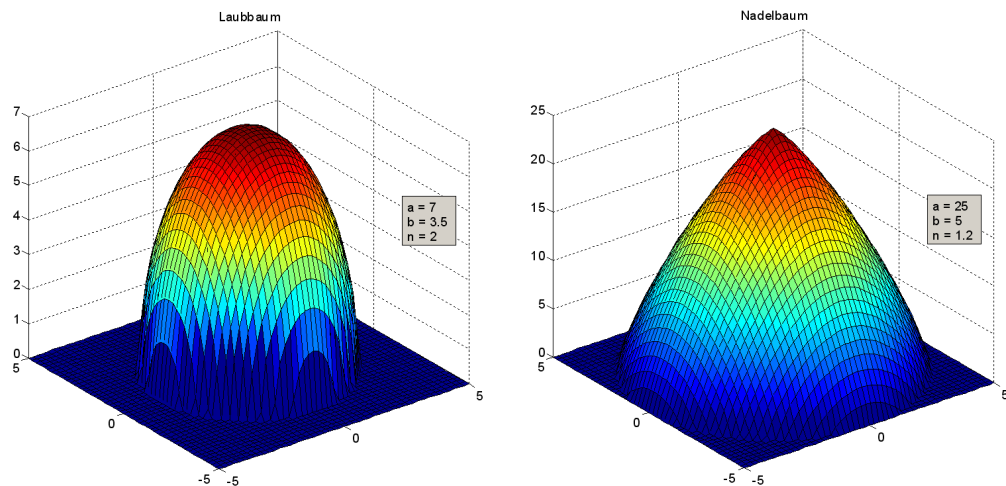


Abbildung 15: 3-D-Visualisierung des Pollock-Modells; Links: Laubbaums; Rechts: Nadelbaum

Die folgende Übersicht beschränkt sich auf einzelbaumbasierte Verfahren, denen als Ausgangsmaterial Höhenmodelle, basierend auf ALS-Daten zugrunde liegen. Alle hier aufgeführten Verfahren, der Methoden und Ergebnisse sind bei Bechtel ausführlich beschrieben und diskutiert (Bechtel 2007). Im Folgenden werden die unterschiedlichen Ansätze, sowie ihre Merkmale tabellarisch zusammengefasst.

Tabelle 1: Morphologischer Ansatz nach Andersen et al.

Merkmale	Ausprägung
Veröffentlichung	(Andersen, Reutebuch & Schreuder 2001)
Modellannahme	Konische oder ellipsoide Baumspitzen
Ausgangsdaten	LIDAR und separates DGM
LIDAR-Auflösung	3,5 Pulse pro m ²
Segmentierung	Morphologische Top-Hat-Transformation
Aggregation	Einzelbaum
Extrahierte Parameter	Baumspitze, Höhe

Tabelle 2: Watershedtransformation im Maßstabsraum nach Straub

Merkmale	Ausprägung
Veröffentlichung	(Straub 2003)
Modellannahme	Größe, Kreisförmigkeit, Konvexität und Vitalität
Ausgangsdaten	LIDAR und radiometrische Daten
LIDAR-Auflösung	verschiedene
Segmentierung	Watershed-Transformation in verschiedenen Maßstabsebenen
Aggregation	Einzelbaum
Extrahierte Parameter	Kronenbegrenzung (Kontur) und Fläche

Tabelle 3: NATSCAN-Projekt von Diedershagen et al.

Merkmale	Ausprägung
Veröffentlichung	(Diedershagen O. 2003)
Modellannahme	Lokale Maxima, Größe und Formparameter (Anisometrie, Kompaktheit), Abstand
Ausgangsdaten	LIDAR, RGB-NIR, FOGIS (Forestal Geographica Information System Baden Württemberg)
LIDAR-Auflösung	1 m
Segmentierung	Pouring-Algorithmus, anschließend Ray-Algorithmus
Aggregation	Bestand und Einzelbaum
Extrahierte Parameter	Bestand: Mittlere Höhe, Kronendichte Einzelbaum: Höhe und Kronenfläche Klassifikation in Laubbaum und Nadelbaum

Tabelle 4: HIGHSCAN nach Hyypä et al.

Merkmale	Ausprägung
Veröffentlichung	(Hyypä et al. 1999), (Hyypä & Inkinen 1999)
Modellannahme	Lokale Intensitätsmaxima (Blobs); Mindesthöhe der Kronenpunkte 6 m
Ausgangsdaten	LIDAR (First Pulse, Last Pulse)
LIDAR-Auflösung	1-20 Pulse pro m ²
Segmentierung	Watershed-Transformation auf einem invertierten Oberflächenmodell
Aggregation	Bestand und Einzelbaum
Extrahierte Parameter	Direkt: Höhe, Kronenbegrenzung und Kronenfläche Indirekt: Stammdurchmesser und Holzvolumen
Anwendungsgebiet	Alpiner Nadelwald; Hauptsächlich Fichte

Tabelle 5: Hill-Climbing nach Persson

Merkmale	Ausprägung
Veröffentlichung	(Heurich et al. 2004)
Modellannahme	Lokale Maxima, Paraboloid (ähnlich dem Pollock-Modell), Höhe des Kronenrandes bei 2m
Ausgangsdaten	LIDAR (First Pulse, Last Pulse)
LIDAR-Auflösung	26-368 cm Footprint, Rasterweite 1/3m
Segmentierung	Hill-Climbing-Algorithmus
Aggregation	Einzelbaum
Extrahierte Parameter	Direkt: Baumspitze/Position, Kronendurchmesser Abgeleitet: Stammdurchmesser, Holzvorrat

2.5 Grundlagen der Parallelen Prozessierung

2.5.1 Parallelen Prozessierung in der Fernerkundung

Die Fortschritte der Sensortechnologien sowohl bei Satelliten- als auch bei flugzeuggetragenen Systemen revolutionieren nicht nur die Datenerfassung in der Fernerkundung. Durch die stetig steigende Menge an Daten und eine wachsende Informationsdichte verändern sich auch das Management, die Speicherung und insbesondere die immer komplexere und rechenaufwändigere Verarbeitung von Fernerkundungsdaten. Die daraus entstehenden Anforderungen an ein Verarbeitungssystem können mit Hilfe neuester Entwicklungen aus dem Bereich des High-Performance-Computing (HPC) erfüllt werden (Plaza 2008). Als Lösungen derartiger Aufgaben mit Hilfe von HPC stellt Plaza neben dem, in der vorliegenden

Arbeit relevanten, verteilten Rechnen weitere Technologien aus dem Bereich massiv paralleler Multiprozessoren sowie hochspezialisierter Hardwarearchitekturen vor (Plaza 2008).

Um im Folgenden auf die Umsetzung des verteilten Rechnens im Rahmen dieser Arbeit hinzuführen, werden zunächst die wichtigsten Begriffe aus dem Bereich der parallelen Verarbeitung geklärt.

2.5.2 Merkmale paralleler und verteilter Systeme

„Die Entwicklung von Computern steht heute vor einem Wendepunkt“ (Bengel et al. 2008). Bengel weist darauf hin, dass nach Jahrzehnten stetiger Steigerung der Rechengeschwindigkeiten je Prozessor nun bereits seit mehreren Jahren ein klarer Trend hin zu Computer-Architekturen erkennbar ist, die parallel Abläufe und die Parallelisierung von Anwendungen unterstützen. „Parallele Architekturen werden zukünftig zum Standard und verlassen damit ihre bisherige Nische des Hochleistungsrechnens“ (Bengel et al. 2008). So bahnt sich derzeit das High Performance Computing (HPC) rasch einen Weg über die Grenzen von Hochschulen und Forschungseinrichtungen hinaus. Schlüsselfertige HPC-Systeme ermöglichen inzwischen jeder Forschungseinrichtung und jedem Unternehmen den Betrieb leistungsstarker Rechnersysteme (Bauke & Mertens 2006).

Parallele Verarbeitung

Ziel der parallelen Verarbeitung ist die Steigerung der Verarbeitungsgeschwindigkeit einer Anwendungen oder eines Programms. Diese Leistungssteigerung erzielt man primär durch das Aufteilen der Applikationen in Einheiten, sogenannten Tasks, oder aus Betriebssystemsicht in Prozesse. Aus Sicht der Softwarearchitektur wird bei einer solchen Aufteilung zwischen nebenläufigen und kooperierenden Prozessen unterschieden. Nebenläufige Prozesse laufen voneinander unabhängig. Kooperierende Prozesse tauchen zur Laufzeit untereinander Informationen aus, um zu einem gemeinsamen Ergebnis zu gelangen.

Im konkreten Fall der Prozessierung von ALS-Daten, die in Kacheln zu je 4 km² (die reorganisierte Prozesskacheln haben eine Fläche von 0,04 km²) organisiert sind, können diese unabhängig voneinander und damit nebenläufig bearbeitet werden. Ein Informationsabgleich zur Laufzeit ist nicht erforderlich. Und das Zusammenfügen der

einzelnen Kachelergebnisse ist nicht Bestandteil der eigentlichen Prozessierung sondern erfolgt unabhängig davon nach Abschluss der Berechnung.

Verteilte Verarbeitung

Ziel der verteilten Verarbeitung bzw. des verteilten Rechnens (*Distributed Computing*) ist die Koordination unterschiedlicher Computer zur Abwicklung einer gemeinsamen Aufgabe. Hardware, Betriebssystem, Programmiersprachen und Ressourcen der einzelnen Computer müssen dabei nicht zwingend einheitlich sein. Wichtig ist primär nur die Vernetzung der Maschinen. Diese basiert in der Regel auf dem TCP/IP-Protokoll. Eine detaillierte und fundierte Einführung in die verteilte Verarbeitung, einschließlich der zugrundeliegenden Netzwerkprotokolle, der unterschiedlichen Ausprägungen von Systemarchitekturen sowie der verschiedenen Jobverteilungsstrategien findet sich bei Bonn (Bonn 2008).

Verteilte versus Parallele Verarbeitung

Verteilte Programme laufen immer auf mehreren Prozessoren, während parallele Programme auch quasiparallel, also auf einem Prozessor ausgeführt werden können. Die verteilte Verarbeitung konzentriert sich somit auf die die Nutzung verschiedener, räumlich verteilter Ressourcen, während bei der parallelen Verarbeitung die Gleichzeitigkeit im Vordergrund steht.

2.5.3 Cluster-Computing

Ein Computercluster, meist einfach Cluster genannt, bezeichnet ein System vernetzter Computer, die von außen als ein Computer gesehen werden können. Die einzelnen Rechner eines Clusters werden Knoten genannt. Die Knoten eines Clusters sind über ein schnelles Netzwerk verbunden. Es gibt unterschiedliche Architekturen und Ausprägungen von Clustern, abhängig vom jeweiligen Anwendungsziel. Eine grobe Unterteilung erfolgt in High-Availability Cluster, Load-Balancing Cluster und High-Performance Cluster (Bengel et al. 2008).

High-Availability Cluster

High-Availability Cluster (*HA-Cluster*) werden zur Steigerung der Verfügbarkeit bzw. für bessere Ausfallsicherheit eingesetzt. Solche Hochverfügbarkeitscluster kommen bei sicherheits- oder geschäftskritischen Anwendungen wie beispielsweise im Energie- und Transportwesen, im Finanzwesen oder in Anlagensteuerungssystemen zum Einsatz.

Load-Balancing Cluster

Load-Balancing Cluster (*Server Load Balancing, SLB*) kommen dort zum Einsatz, wo mit hohen Netzwerk- oder Rechnerlasten gerechnet wird. Die Architektur solcher Systeme ermöglicht es, einen sukzessiv steigenden Leistungsbedarf durch das Hinzufügen zusätzlicher Rechner zu kompensieren. Grund für die Verwendung ist nicht zuletzt der Einsatz von preisgünstigen Standardcomputern (*Commercial off-the-shelf, COTS-Komponenten*) anstatt teurer Spezialcomputer. Solche Systeme finden sich typischerweise im Umfeld von Internet-Diensteanbietern, um deren Rechenkapazitäten flexibel auf steigende Zugriffszahlen anpassen zu können.

High-Performance-Computing Cluster

High-Performance-Computing Cluster (*HPC-Cluster*) dienen zur Abarbeitung von umfangreichen und zeitintensiven Rechenaufgaben. Dabei wird eine Rechenaufgabe in mehrere Pakete, auch Jobs genannt, aufgeteilt und anschließend parallel auf den Knoten des Clusters ausgeführt. Die Auf- und Verteilung der Jobs übernimmt eine Job Management Software. HPC-Cluster finden sich häufig im wissenschaftlichen Bereich.

High-Performance-Computing Cluster BEOWULF

Neben einer Vielzahl kommerzieller Clustersysteme (IBM, Sun, Oracle, HP, Microsoft) hat sich insbesondere im Linux-Umfeld der Beowulf-Cluster etabliert. Der Name geht zurück auf das von Sterling und Becker 1994 im Auftrag der NASA aufgebaute System aus 16 handelsüblichen Linux-PCs (Sterling et al. 1995). Heute gibt es kaum eine Universität oder Forschungseinrichtung, die nicht mindestens einen Beowulf-Cluster betreibt (Bengel et al. 2008).

In einem Beowulf-Cluster werden mehrere Computer zu einem Netzwerk zusammengeschlossen. Die Knoten (Arbeitsrechner) übernehmen die eigentliche Aufgabe des Clusters. Die Steuerung des Beowulf-Clusters erfolgt über einen Masterknoten. Dieser übernimmt auch die Aufgabenverteilung.

Die Realisierung eines BEOWULF-Clusters kam im Rahmen der vorliegenden Arbeit nicht in Frage, da die erwartenden Rechenkapazitäten bei dem beteiligten Kooperationspartner (KMU aus dem Bereich Fernerkundung) nicht kontinuierlich sondern nur projekt- und auftragsabhängig anfallen. Die Investition in die eigens für den Aufbau des Clusters notwendige Hardwareausstattung, auch wenn es sich nur um

COTS-Komponenten handelt, war zum gegebenen Zeitpunkt wirtschaftlich nicht rentabel.

2.5.4 Grid-Computing

Cluster-Computing versus Grid-Computing

Was mit **Cluster**-Computing erreicht werden will, kann auf einen Nenner gebracht werden: Leistungssteigerung durch das Vernetzen von Rechnern.

Beim **Grid**-Computing besteht der zentrale Gedanken darin, einem größeren Kreis von Anwendern Zugriff auf verteilte Ressourcen zu verschaffen. Grid-Computing ist eine Technik zur Integration organisationsübergreifend und ortsunabhängig verteilter Ressourcen auf Basis bestehender Kommunikationsinfrastrukturen wie der des Internets. Die gemeinsame Nutzung der verteilten Ressourcen erfolgt in virtuellen Organisationen (VO), in denen die Ressourcen den Benutzern des Grids dynamisch zugeordnet werden.

Das Hauptunterscheidungsmerkmal von Grid-Computing und Cluster-Computing besteht darin, dass die Ressourcen in einem Grid üblicherweise verschiedenen, unabhängigen Organisationen gehören. Die Betriebsmittel eines Clusters hingegen gehören in der Regel nur zu einer einzigen administrativen Einheit und werden von dieser betrieben und verwaltet.

Die dem Grid-Computing zugrunde liegende Architektur wurde unter anderem von Ian Foster entworfen, der das Prinzip der „Open Grid Service Architecture“ (OGSA) veröffentlichte (Foster & Kesselman 1999). Mit dem Open Grid Forum (OGF, Open Grid Forum 2009) wurde eine Standardisierungsgremium für Grid-Computing geschaffen, in dem Vertreter aus Wissenschaft und Industrie zusammenarbeiten.

Grid-Middleware

Die essentielle Komponente eines Grid ist die Grid-Middleware (*deutsch etwa „Zwischenanwendung“*), eine Software, mit der der Anwender einfach, sicher und effizient auf die Rechenleistung, die verteilten Daten und angebundene Sensorsysteme zugreifen kann. Im Gegensatz zu niveautieferen Netzwerkdiensten wie TCP/IP, welche die einfache Kommunikation zwischen Rechnern handhaben, unterstützt die Middleware die Kommunikation zwischen Anwendungen und Diensten. Abbildung 16 veranschaulicht die unterschiedlichen Schnittstellen einer Grid-Infrastruktur und

verdeutlicht somit die Notwendigkeit und die vielseitigen Aufgaben einer Grid-Middleware.

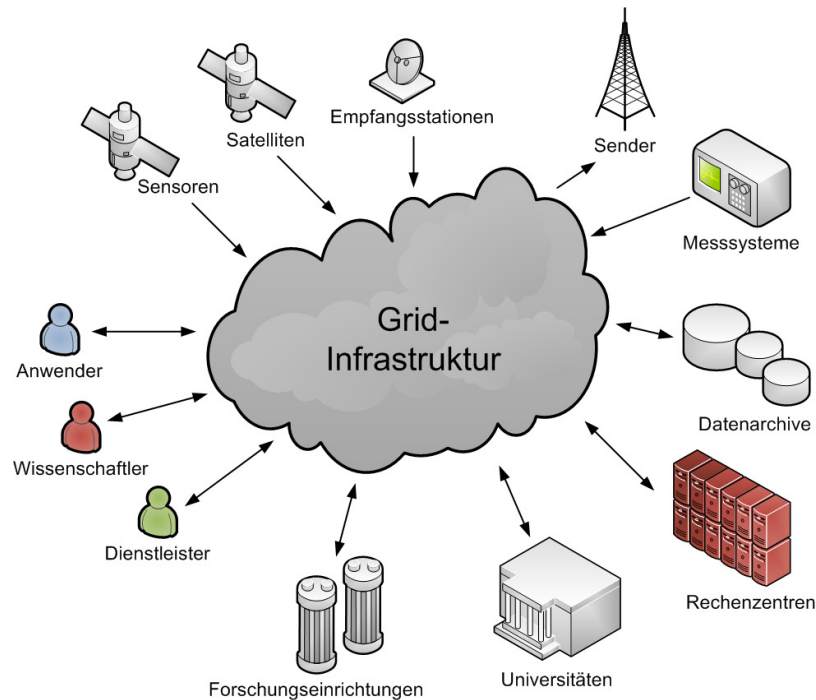


Abbildung 16: Grid-Infrastruktur mit Schnittstellen zu Sensor- und Datenverarbeitungssystemen

Inzwischen haben sich im Bereich des Grid-Computing verschiedene Middleware-Systeme mit unterschiedlichen Schwerpunkten etabliert. Die am weitesten Verbreiteten Systeme sind Globus Toolkit 4 (GT4), gLite 3.0 (EGEE, basiert auf GT2) und Unicore. Neben einer grundlegenden Einführung in diese Middleware-Systeme bei Bengel (Bengel et al. 2008) bietet insbesondere der Sammelband des Projekts D-Grid (Neuroth, Kerzel & Gentsch 2007) eine gute Übersicht über praxisorientierte Anwendungen der unterschiedlichen Systeme.

Grid-Projekte in Fernerkundung und Geoinformatik

Cai et al. (Bubak & Albada 2004) führen bereits 2004 Untersuchungen mit Grid-Technologien im Rahmen der Prozessierung von Fernerkundungsdaten durch und kommen zu dem Schluss, dass der Einsatz von Grid Computing System machbar ist. Das Projekt GDI-Grid (Neuroth, Kerzel & Gentsch 2007) befasst sich mit der effizienten Erschließung und Prozessierung von Geodaten für die Geosimulation von Lärmausbreitung und Katastrophenfällen und ist Teil der vom BMBF initiierten D-Grid-Initiative, einem Verbundprojekt mit mehr als 150 Einzelprojekten. Die Ziele des Projekts bestehen darin, die in den GDI bisher ungenügend abgedeckten Anforderungen an elektronische Ressourcen mittels Grid-Architekturen und -Diensten zu erfüllen und

Grid-fähige Dienstketten auf der Basis von OWS (OGC Web Services) konformen aufzubauen. Weitere Arbeiten zur Anwendung von Grid-Technologien in der Fernerkundung und Geoinformatik finden sich bei Plaza (Plaza 2008) und Slood (Slood 2005).

2.5.5 Desktop Grid-Computing

Das Konzept des Grid-Computing besteht in der Vernetzung organisationsübergreifender Ressourcen um diese einer breiteren Anwenderschicht zur Verfügung zu stellen (siehe Abbildung 16). Beim Desktop Grid-Computing wendet man dieses Konzept nicht auf organisationsübergreifende Ressourcen an, sondern nur auf die innerhalb einer Organisation bereits vorhandenen und vernetzten Rechenkapazitäten. Ziel ist die kostengünstige Bereitstellung von Rechenleistung bereits vorhandener Arbeitsplatzrechner. Auf die Anschaffung zusätzlicher Hardware zum Aufbau eines Rechenclusters kann verzichtet werden. Ebenso ist die Vernetzung der Rechner bereits gegeben.

2.6 Das System BOINC

2.6.1 Die Idee

Die Idee des Cluster-Computings ist es, Rechner mit der Absicht zu vernetzen, einen leistungsstarken Parallelrechner zu betreiben. Für diesen Zweck müssen die, zwar kostengünstigen, dennoch zweckgebundenen Hardwareressourcen jedoch erst beschafft und das Cluster eingerichtet werden.

Wendet man diese Idee des Cluster-Computing nun aber auf bereits bestehende Rechnernetzwerke wie Internet oder Intranet an, so kann eine Synergie erzielt werden, die für beide Seiten gewinnbringend ist. Seitens des Cluster-Computing müssen keine eigenen Hardwareressourcen beschafft werden. Auch die Vernetzung der Rechner ist bereits gegeben. Auf Seiten der bestehenden Infrastruktur führt die Nutzung von Leerlaufzeiten der Rechner zu einer höheren und damit wirtschaftlicheren Ausschöpfung der Ressourcenkapazitäten.

David Gedye und Craig Kasnoff von der University of California at Berkeley machten sich 1996 diese Idee des verteilten Rechnens via Internet zu Nutze und begannen mit der Entwicklung des Programms SETI@Home (Anderson et al. 2002). SETI steht dabei für „*Search for Extraterrestrial Intelligence*“. Das Programm lief von 1999 bis 2006. In

dieser Zeitspanne waren insgesamt über 5 Millionen Computer mit einer Gesamtrechenzeit von mehr als 2,4 Millionen Jahre an dem Projekt beteiligt.

Das Konzept der Nutzung bestehender PC-Ressourcen wurde von weiteren Projekten aufgegriffen, welche sich mit unterschiedlichen naturwissenschaftlichen Problemen beschäftigten. In der Folge entstand aus diesem großen Interesse an der zugrundeliegenden Architektur das System BOINC (Berkeley Open Infrastructure for Network Computing, University of California 2009), eine allgemeine Softwareplattform für das verteilte Rechnen in offenen Netzwerken.

BOINC ermöglicht die Nutzung freier Rechenleistungen von Computern über das Internet oder Intranet. Ursprünglich für wissenschaftliche Berechnungen entwickelt, wird BOINC inzwischen in vielen Bereichen eingesetzt. Seit 2003 steht BOINC unter der GNU General Public License (GPL) und ist somit für alle Interessenten, die eigene Rechen-Projekte auf das Basis von BOINC entwickeln wollen, offen zugänglich.

2.6.2 Die Architektur von BOINC

Serverseitige Komponenten

Das System basiert auf einer Client-Serverarchitektur. Der Server kann mittels Konfiguration flexibel an unterschiedliche Projektanforderungen angepasst werden. Es können verschiedene Projektanwendung (Rechenprogramme) und zugehörige Arbeitspaket-Sammlungen definiert werden. Aufgabe des Servers ist die Verwaltung von Projekten und Benutzern, die Verteilung von Arbeitspaketen und das Einsammeln der Berechnungsergebnisse. Dazu müssen einerseits verschiedene systemspezifische Dienste konfiguriert und andererseits projektspezifische Dienste implementiert werden. Auf der Basis mitgelieferter Beispielprojekte können einfache Projekte zügig eingerichtet werden.

Das Backend besteht insgesamt aus einem Webserver, PHP als Skriptsprache und einer MySQL-Datenbank. Im Rahmen der vorliegenden Arbeit wurde ein vorinstalliertes Komplettsystem in Form einer Virtual Machine eingesetzt. Die Virtual Machine kann von der Webseite des Projekts BOINC heruntergeladen werden (University of California 2009).

Abbildung 17 skizziert in einem Komponentendiagramm die einzelnen Komponenten des BOINC-Systems und deren Verteilung auf Client und Server. Die rot markierten Komponenten müssen auf die individuellen Anforderungen eines spezifischen BOINC-

Projekts abgestimmt werden. Die blauen Komponenten sind BOINC-Systemkomponenten und unabhängig vom spezifischen Projekt.

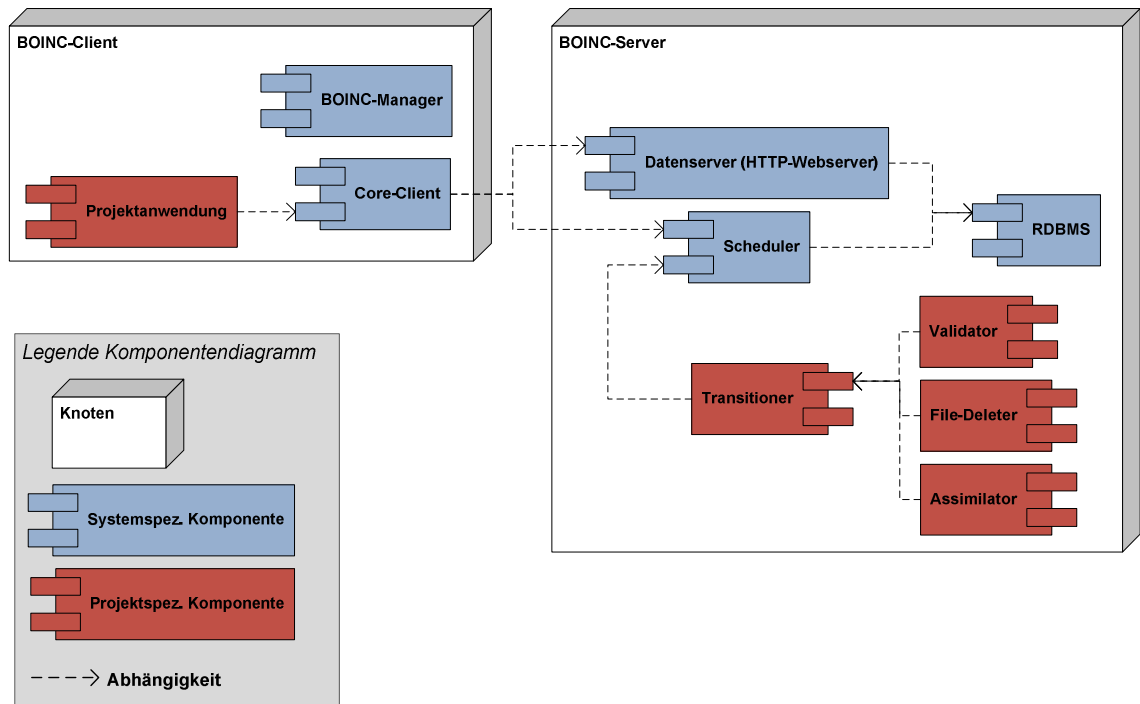


Abbildung 17: Komponenten der BOINC-Plattform

Im Folgenden wird die Funktion der einzelnen Serverkomponenten im Detail beschrieben:

Der *Scheduler* ist das zentrale Steuerungsprogramm des BOINC-Servers. Er verteilt Arbeitspakete an die Clients und verarbeitet deren Ergebnismeldungen. Alle Aktivitäten des Schedulers werden in der Datenbank dokumentiert.

Ein einfacher HTTP-Server dient als *Datenserver*, von dem der Client das zugeteilte Arbeitspaket herunterladen und die Ergebnisdateien hochladen kann.

Der *Validator*, ein für jedes Projekt unterschiedliches Programm, prüft die vom Client zurückgelieferten Ergebnisdateien auf Korrektheit. Meist geschieht dies dadurch, dass ein Arbeitspaket von mehreren Teilnehmern redundant bearbeitet wird. Der Validator vergleicht dann die Ergebnisse. Idealerweise sind sie identisch.

Der *Assimilator* ist ebenfalls projektspezifisch zu implementieren. Er nimmt die validierten Ergebnisdateien und bereitet sie zur weitergehenden wissenschaftlichen Analyse auf. So können die Ergebnisse beispielsweise in eine weitere, projektspezifische Datenbank übertragen werden.

Nachdem die Ergebnisse „assimiliert“ wurden, sind die Input- und Output-Dateien der Clients unnötiger Ballast für den Datenserver, die durch ihre Anzahl auch die Performance beeinträchtigen können. Mit dem *File-Deleter* werden nicht mehr benötigte Dateien vom Server gelöscht.

Der *Transitioner* überwacht den Fortschritt der Arbeitspakete entlang einer gedachten Pipeline. So stößt er beispielsweise den Validator an, wenn er feststellt, dass zu einem Arbeitspaket genügend redundante Ergebnisse vorliegen, so dass mit der Validierung begonnen werden kann.

In der vorliegenden Arbeit wird nur ein rudimentäres BOINC-Projekt aufgesetzt. Auf eine projektspezifische Implementierung eines Validators und eines Assimilator konnte dabei verzichtet werden.

Clientseitige Komponenten

Auf den Client-Rechnern wird der BOINC-Standardclient installiert. Der Standardclient besteht auf dem Core-Client sowie dem BOINC-Manager.

Der *Core-Client* wird auf dem Knoten-Rechner als Dienst installiert und läuft ohne graphische Oberfläche. Er steuert und überwacht die Projektanwendungen gemäß der Rechnerkonfiguration, puffert Arbeitspakete und kommuniziert mit dem Scheduler und Datenserver des Projekts. Der Core-Client steht sowohl für Windows-, als auch für Unix- Plattformen zur Verfügung.

Der *BOINC-Manager* bietet dem Nutzer eine grafische Oberfläche zur Konfiguration und Überwachung des Core-Clients.

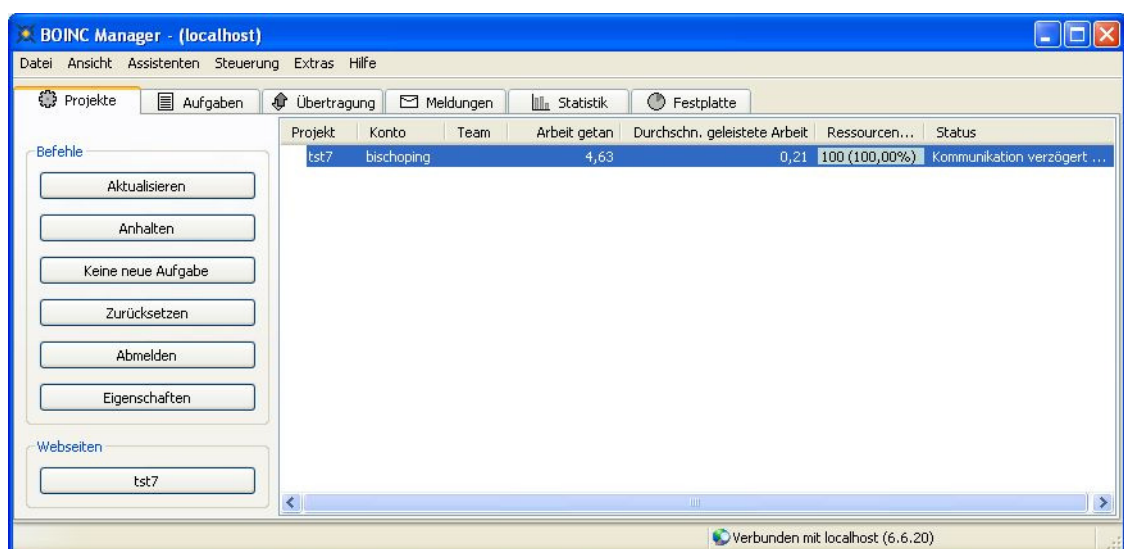


Abbildung 18: Screenshot des Boinc-Manager

Das **BOINC-Projekt** stellt eine oder mehrere Anwendungen bereit, die vom Core-Client heruntergeladen und zur Berechnung der Arbeitspakete verwendet werden. Die Arbeitspakete werden vom Core-Client überwacht. Die Funktionen für diese Überwachung werden in der von BOINC mitgelieferten Programmierschnittstelle (BOINC-API) bereitgestellt.

2.6.3 Wrapper-Konfiguration

In der Regel muss eine Projektanwendung die BOINC-API einbinden, damit der Core-Client mit der Anwendung kommunizieren und diese steuern kann. Es gibt jedoch eine Möglichkeit, auch bestehende Anwendungen ohne Änderung des Programmcodes als BOINC-Projekt ausführen zu lassen. Dazu wird von BOINC eine sogenannte Wrapper-Projektanwendung einschließlich einer Wrapper-Projektanwendung bereitgestellt. Diese Wrapper-Anwendung kapselt die eigentliche Projektanwendung. Der Wrapper übernimmt die Kommunikation mit dem Core-Client und kann von diesem gesteuert werden.

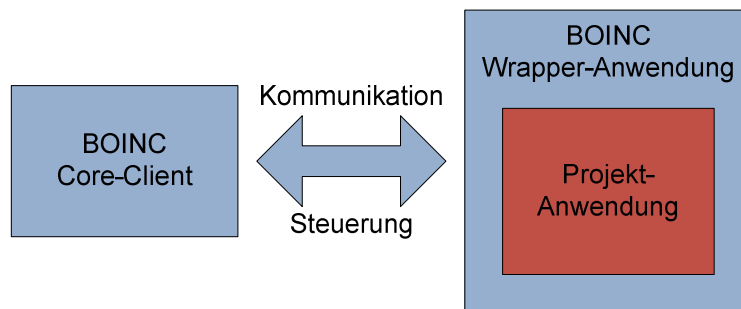


Abbildung 19: Kapselung der Projektanwendung durch die Wrapper-Anwendung

Die Variante der Projektkonfiguration mit Hilfe eines Wrappers wurde auch in der vorliegenden Arbeit eingesetzt, da es hier weniger um die Details eines vollständigen BOINC-Projekts geht, sondern vielmehr um die Untersuchung der Anwendbarkeit der BOINC-Plattform auf umfangreiche Bildverarbeitungsaufgaben.

2.6.4 Job-Verteilverfahren

Die Verteilung der Jobs im BOINC-System wird von den Rechenknoten gesteuert. Der auf dem Rechenknoten installierte Core-Client entscheidet anhand der vom Benutzer eingestellten rechner-spezifischen Konfiguration und der aktuellen Rechnerauslastung, wann er arbeiten möchte und wie viele lokale Ressourcen für die Bearbeitung von Jobs verfügbar sind bzw. zugeteilt werden können. Es findet also kein aktives Jobscheduling auf seitens des Servers statt.

3 Material und Untersuchungsgebiet

3.1 Untersuchungsgebiet, Testdaten und Referenzflächen

3.1.1 Untersuchungsgebiet

Das Untersuchungsgebiet umfasst die gesamte Fläche des Bundeslandes Baden-Württemberg. Von Seiten des Projektpartners wurden die Daten in 9311 Kacheln à 2 km x 2 km geliefert. Dies ergibt eine zu untersuchende Fläche von etwa 37.000 km².

Die Daten sind im Vorfeld vom Projektpartner vorverarbeitet worden. Dabei sind Bereiche, in denen großflächig keine Streuobstflächen enthalten sind, wie beispielsweise Städte, vollständig herausgefiltert worden. Desweiteren standen einzelne Ausschnitte der Daten bis zum Projektstart nicht zur Verfügung. Fehlende oder unvollständige Kacheln sind in der folgenden Abbildung hervorgehoben.

Untersuchungsgebiet Baden-Württemberg: 9311 Kacheln (2 km x 2 km)

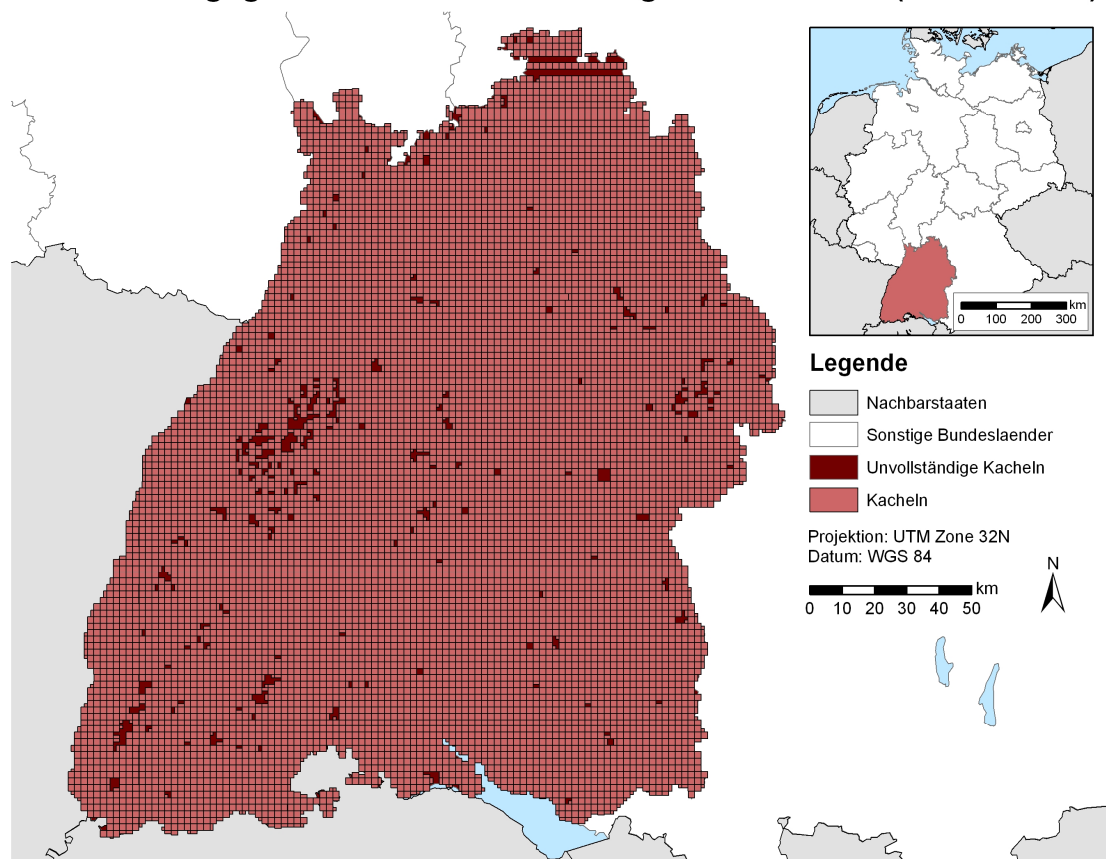


Abbildung 20: Kachelung des Untersuchungsgebietes

Zu jeder Kachel wurden vom Projektpartner die ALS-Daten (siehe Kapitel 3.2), sowie die Masken (siehe Kapitel 3.3) geliefert. Weiterhin wurden vom Projektpartner für

einzelne charakteristische Gebiete Orthophotos mit einer Auflösung von 25 cm bereitgestellt (siehe Kapitel 3.1.4).

3.1.2 Testdaten für die Entwicklung des Algorithmus

Bei der Entwicklung des Algorithmus zur Einzelbaumextraktion wurden verschiedene, charakteristische Teilgebiete von je 200 m x 200 m ausgewählt. Es wurden nur Gebiete verwendet, für die auch Orthophotos zur Verfügung standen. Bei der Auswahl der Testdaten wurden folgende, charakteristische Merkmale berücksichtigt:

- Unterschiedliche Größe und Abstände von Einzelbäumen
- Baumgruppen
- Hochspannungsleitungen und Strommasten
- Waldränder
- Nicht sauber abgegrenzte Masken aufgrund veralteter Datengrundlagen („nichtmaskierte Gebäude“)

3.1.3 Referenzflächen für die Validierung

Da bis zur Abschlussphase der vorliegenden Arbeit vom Projektpartner keine Ground-Truth Daten bereitgestellt wurden, erfolgt die Validierung des Algorithmus zur Einzelbaumextraktion auf der Grundlage der verfügbaren Orthophotos.

Auswahl der Referenzflächen

Um das Verhalten und die Qualität des Algorithmus zu verifizieren, wurden zu jedem der in Kapitel 3.1.2 beschriebenen charakteristischen Merkmale ein oder zwei Referenzflächen (Prozesskacheln) mit einer Fläche von je 200 m x 200 m anhand der Orthophotos ausgewählt. Bedingung bei der Auswahl der Referenzflächen war, dass diese nicht mit den Testdaten der Entwicklung übereinstimmen dürfen.

Als Vorbereitung des Accuracy Assessments (siehe Kapitel 2.3.6) wurden alle in den Orthophotos der Referenzflächen erkennbaren Streuobstbäume von einem unabhängigen Experten digitalisiert.

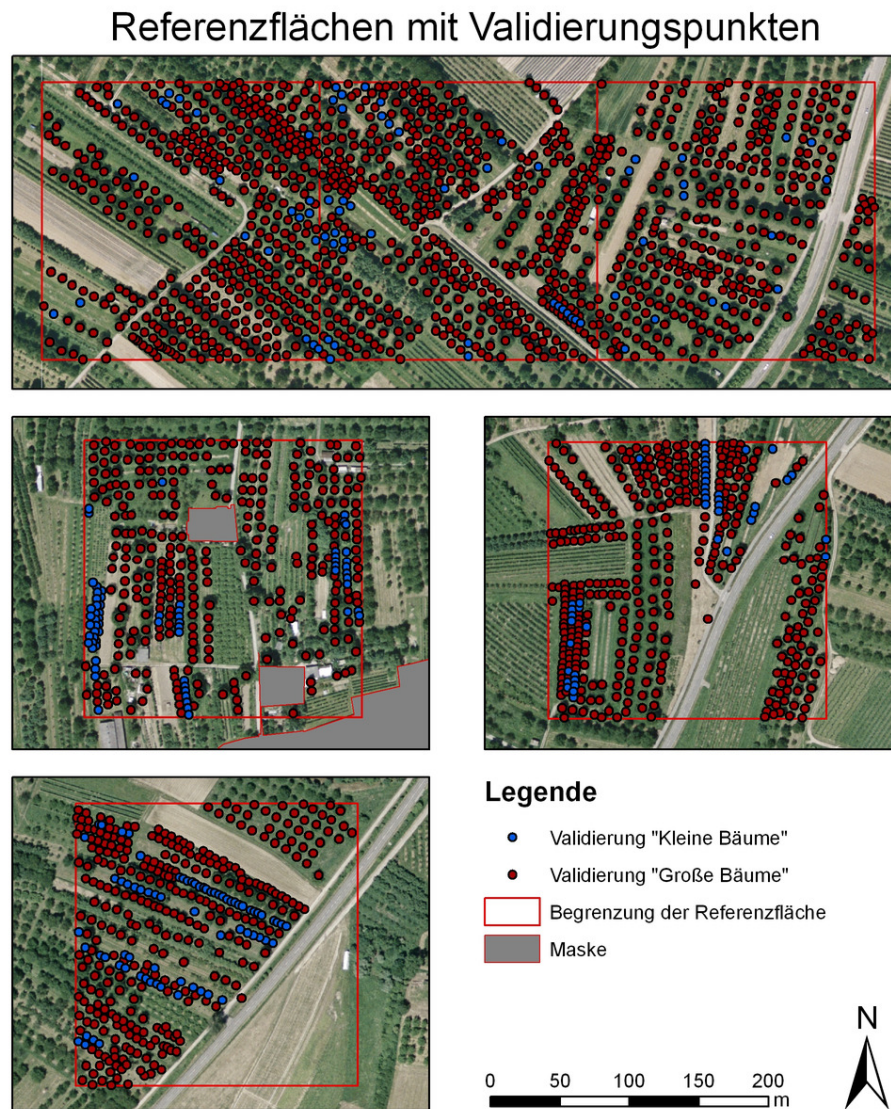


Abbildung 21: Auswahl der Referenzflächen

3.1.4 Zusätzliche Daten für Entwicklung und Validierung

Zur Unterstützung der Entwicklung des Algorithmus sowie zur Validierung wurden digitale Orthophotos (DOP) mit einer Bodenauflösung von 25 cm aus verschiedenen Befliegungen des Landesvermessungsamtes Baden-Württemberg verwendet. Die Orthophotos stehen im GEO-TIFF-Format bereit und haben eine Größe von 4000 x 4000 Pixeln (1 km x 1 km).

Sowohl die Orthophotos als auch die ALS-Daten stammen aus unterschiedlichen Befliegungskampagnen, die teilweise mehrere Jahre auseinander liegen. Die Daten weisen daher inhaltlich teilweise beträchtliche Unterschiede auf. Abbildung 22 verdeutlicht diesen zeitlichen Unterschied zwischen ALS-Daten und Orthophotos anhand eines Ausschnitts. Im Orthophoto (Abbildung 22, oben) ist deutlich ein Feld erkennbar. Die zugehörigen ALS-Daten weisen an dieser Stelle jedoch First pulse

Reflexionen auf, die auf Vegetation deuten (Abbildung 22, oben rechts). Die unteren Bilder der Abbildung zeigen den gleichen Ausschnitt, allerdings mit einem Screenshot aus Google Earth aus dem Jahr 2000. Die Übereinstimmung zwischen ALS-Daten und Farbfoto ist augenscheinlich. Die ALS-Daten des Ausschnitts sind aus der Befliegungskampagne von 2001. Das entsprechende Orthophoto stammt vom 26.05.2005.



Abbildung 22: Verdeutlichung der zeitlichen Differenz von ALS-Daten und Orthophoto (oben) bzw. Google Earth (unten)

3.2 Laserscan-Daten des Landesvermessungsamts Baden-Württemberg

3.2.1 Befliegungskampagne

Das Landesvermessungsamt Baden-Württemberg hat im Jahr 1999 eine Befliegungskampagne zur Erstellung eines landesweit einheitlichen DGMs initiiert. Als Technik wurde das Airborne Laserscanning (ALS) gewählt (siehe Kapitel 2.1). Die Datenaufbereitung wurde im Jahr 2008 abgeschlossen (Schleyer 2001, Klauser 2008).

Die folgende Abbildung (Abbildung 21) ist den Vortragsunterlagen von Klauser (Klauser 2008) entnommen und veranschaulicht die Planungsgrundlage der Befliegungskampagne.

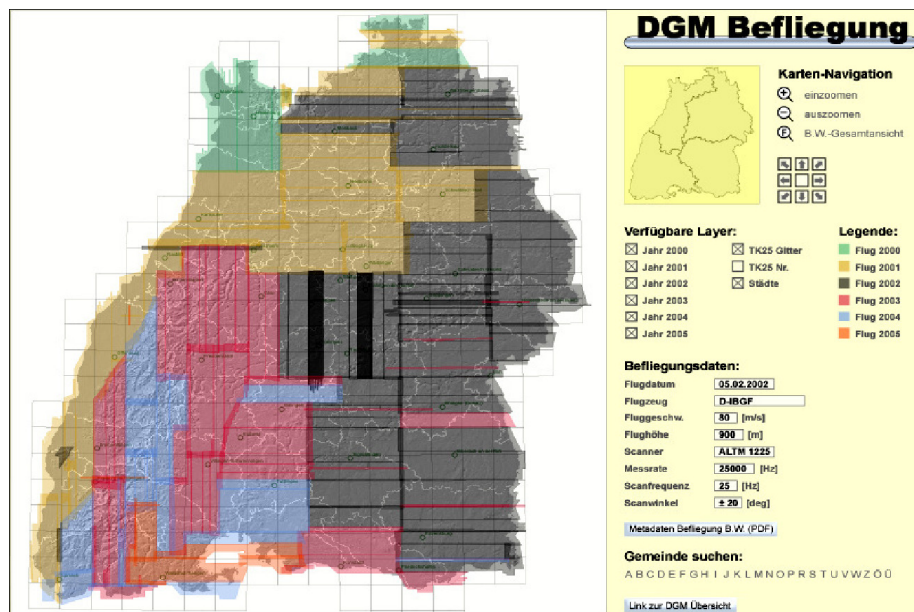


Abbildung 23: Skizze der Befliegungskampagne aus (Screenshot aus (Klauser 2008))

3.2.2 Eckdaten und Parameter der ALS-Datenerfassung

Die gesamte Landesfläche Baden-Württembergs (ca. 35.800 km²) wurde flächendeckend erfasst. Die Befliegungen fanden zwischen 2000 und 2005 in den **vegetationsarmen Wintermonaten Dezember bis April statt**. Der mittlere Abstand der aufgenommenen Punkte beträgt 1 m - 2 m. Die Höhengenaugigkeit der aufgenommenen Punkte liegt bei ± 15 cm (einfache Standardabweichung), d.h. 97 % aller aufgenommenen Punkte stimmen innerhalb von max. ± 50 cm mit den tatsächlichen Höhen überein. Die Georeferenzierung erfolgte im geodätischen Bezugssystem Europäische Terrestrische Referenzsystem 1989 (ETRS89), das Projektionssystem ist Universal Transverse Mercator (UTM). Die erste und letzte Reflexion (first pulse und last pulse) der aufgenommenen Punkte wurden automatisch in Boden- und Hochpunkte klassifiziert (vgl. Kapitel 2.1.1). Anschließend wurden die Daten in Kacheln von 1 km x 1 km in das Gauß-Krüger-Koordinatensystem transformiert.

Bei der Datenerfassung wurde der Airborne Laser Terrain Mapper (ALTM 1225) der kanadischen Firma Optech Inc. eingesetzt. Die Eckdaten des Sensors sind:

- Messfrequenz: 25.000 Hz
- Scanfrequenz: 25 Hz
- Scanwinkel: $\pm 20^\circ$

- Fluggeschwindigkeit 80 m/sec
- Mittlere Flughöhe: 900 m
- Streifenabstand: 300 m – 450 m in Abhängigkeit von der Reliefenergie

Aus den Systemeinstellungen ergibt sich ein mittlerer Punktabstand aller Laserpunkte von 1,5 m.

3.2.3 Genauigkeit der Laserscandaten

Die Genauigkeitsanforderungen seitens Landesvermessungsamt Baden-Württemberg wurden unter anderem durch Sixel, Straub und Wagelaar (Sixel, Straub & Wagelaar 2005) im Rahmen einer Machbarkeitsstudie anhand terrestrischer Vegetationshöhenmessungen verifiziert. Dabei wurden auffällige Offenlandgehölze, wie beispielsweise Apfelbäume einer Streuobstwiese, oder deutlich abgrenzbare Baumkronen innerhalb einer Hecke, terrestrisch vermessen. Insgesamt wurden 54 Messungen verglichen. Der Mittelwert der Messhöhendifferenz aller gemessenen Bäume beträgt $-3,65$ m, bei einer Standardabweichung von $\pm 1,8$ m. Das bedeutet, dass die mit dem Laser ermittelten Baumkronen um ca. 3,6 m geringer ausfallen, als die terrestrisch gemessenen. Die Vegetationshöhen werden im Mittel um 25 % unterschätzt. Diese systematische Abweichung zeigt, dass die Reflexionen bei einer Winterbefliegung häufig nicht von den Wipfeltrieben erfolgen, sondern meist von tiefer gelegenen Ästen. Weiterhin ist die Punktdichte nicht hoch genug, um alle Wipfeltriebe zu erfassen.

3.2.4 Nachbearbeitung der Rohdaten

Die Rohdaten (ca. 37 Milliarden Messpunkte) wurden vom Landesvermessungsamt Baden-Württemberg in einem halbautomatischen Prozess nachbearbeitet. Anschließend wurden verschiedene Produkte, unter anderem das „DGM1“, Digitales Geländemodell mit 1 m Rasterweite, erstellt (Schleyer 2001). Die landesweite Fertigstellung des Geländemodells ist seit Ende 2008 abgeschlossen. Seither werden die Daten des DGM aufgrund permanenter Veränderungen (beispielsweise Baumaßnahmen) regelmäßig aktualisiert. Abbildung 24 stellt den Ablauf des Nachbearbeitungsprozesses dar.

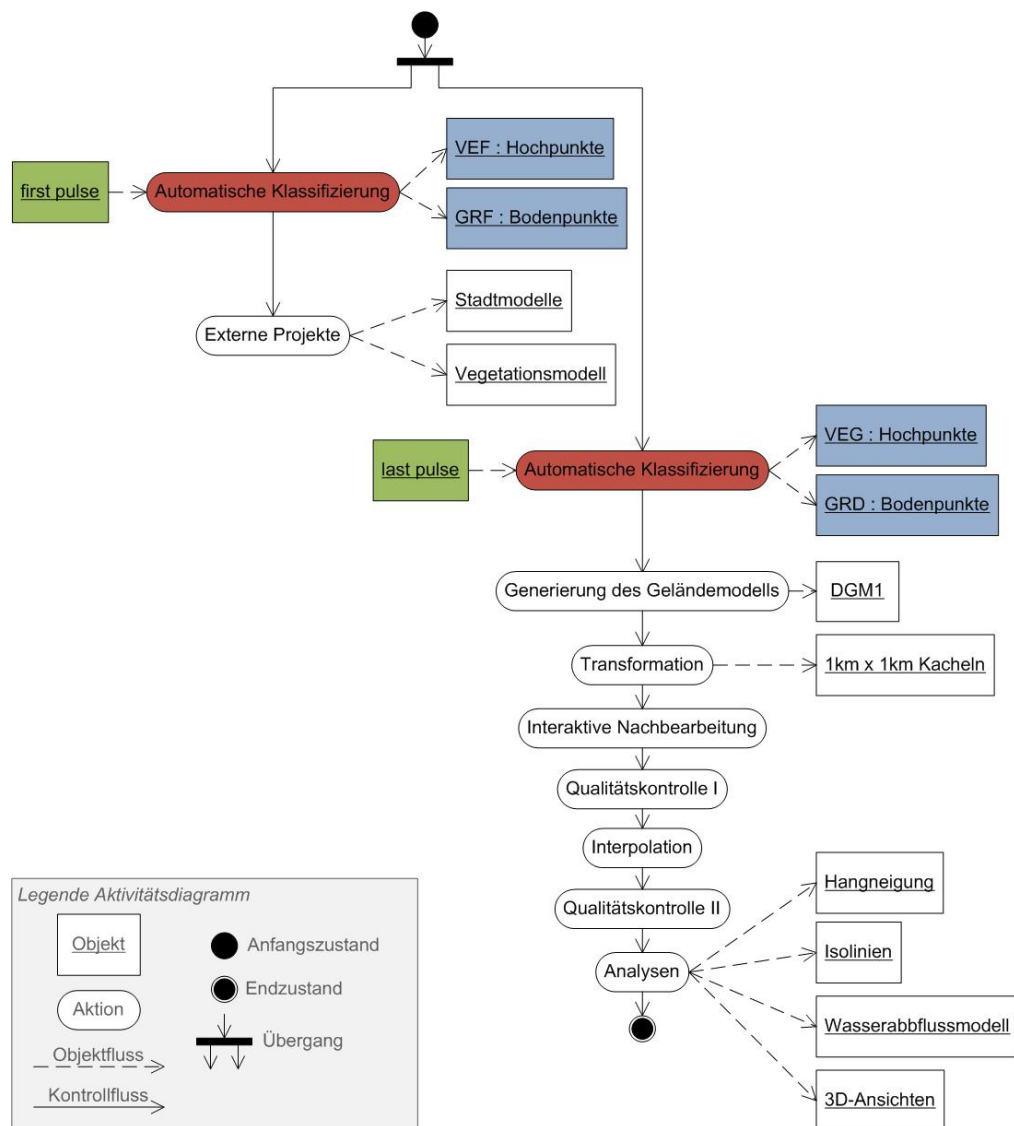


Abbildung 24: Arbeitsschritte zur Erstellung des digitalen Geländemodells (vgl. Schleyer 2001)

In Abbildung 24 sind die Produkte „Hochpunkte“ und „Bodenpunkte“ der automatischen Klassifizierung blau hervorgehoben, da nur diese im weiteren Verlauf der Arbeit von Interesse sind.

Die nachfolgenden Arbeitsschritte konzentrieren sich im Wesentlichen auf die Korrektur von Fehlern und Störungen antropogener Herkunft, wie in der folgenden Zusammenstellung verschiedener Fehlerquellen veranschaulicht wird (Abbildung 25). Die Beispiele sind der Arbeit von Klauser (Klauser 2008) entnommen. Für die vorliegende Arbeit sind diese Nachbearbeitungsschritte nicht relevant, da Datenfehler im Offenland nur minimal sind und somit für die weiteren Betrachtungen vernachlässigt werden können.

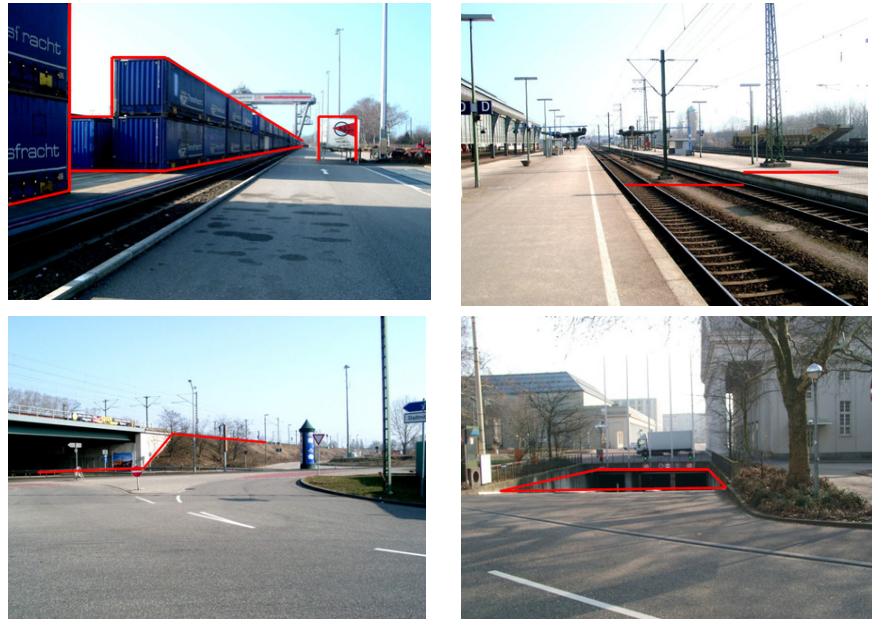


Abbildung 25: Beispiele von Fehlerquellen bei der Generierung des DGM (Klauser 2008)

3.3 Abgrenzung des Kontext

3.3.1 Kontextmodell

Das Kontextmodell dient der Trennung von Gebieten, in denen unterschiedliche Ausprägungen des Objektmodells angewendet werden (Hinze 2004). Es kann aber auch Grundlage einer Segmentierung sein, in der die unterschiedlichen Kontextgebiete generiert werden (Baumgartner 1999).

Grundsätzlich wird zwischen einem lokalen und einem globalen Kontext unterschieden. Während der lokale Kontext Beziehungen zwischen einzelnen Objekten beschreibt, z.B. die Abgrenzung von Bäumen aufgrund der Merkmale des Baummodells, unterteilt der globale Kontext das Bild in Gebiete, in denen das Erscheinungsbild des Objekts vollkommen unterschiedlich geartet sein kann. Dazu gehörten auch Gebiete, in denen das Objekt nicht auftreten darf oder auftreten kann (beispielsweise ein Streuobstbaum in einem Industriegebiet).

Der globale Kontext dient somit als Maske, in dem unterschiedliche Ausprägungen des Objektmodells angewendet werden. Im konkreten Fall der Baumextraktion wird der globale Kontext als Maske (vgl. Kapitel 3.2) verwendet, außerhalb derer das Objektmodell nicht anwendbar ist.

Grundlage dieser Maske ist das normalisierte digitale Oberflächenmodell sowie Daten der Automatisierten Liegenschaftskarte (ALK) und dem Amtlichen Topografisch-Kartografischen Informationssystem (ATKIS).

3.3.2 Normalisiertes Digitales Oberflächenmodell

Das normalisierte digitale Oberflächenmodell (nDOM) ist die Differenz aus dem Oberflächenmodell und dem Geländemodell. Für die Erstellung der Streuobst-Maske wurde ein nDOM auf der Grundlage der automatisch klassifizierten Hoch- und Bodenkpunkte der ALS-Daten (first pulse und last pulse) des Landesvermessungsamts Baden-Württemberg (vgl. Kapitel 3.1.4) erstellt.

3.3.3 Automatisierte Liegenschaftskarte (ALK)

Die Automatisierte Liegenschaftskarte (ALK) bildet gemeinsam mit dem Automatischen Liegenschaftsbuch (ALB) das Liegenschaftskataster. In der ALK werden Punkt- und Grundrissdaten in digitaler Form geführt. In den hier relevanten ALK-Grundrissdaten werden Flurstücke, politische Grenzen, Gebäude, tatsächliche Nutzungen, topographische Objekte, Texte und Punktnummern zur Beschriftung sowie nicht nummerierte Punkte gespeichert.

Die Daten sind objekt- und vektorstrukturiert, maßstabs- und blattschnittfrei und werden kontinuierlich fortgeführt. Die Lagegenauigkeit der Objekte liegt im Zentimeter- und Dezimeterbereich (Landesvermessungsamt Baden-Württemberg 2009b).

3.3.4 ATKIS Basis DLM

Das ATKIS®-Basis-DLM ist ein digitales Landschaftsmodell, das die Landschaft in Form von topographischen Objekten beschreibt. Die Objekte werden einer Objektart zugeordnet (wie z.B. Wald-, Acker- oder Siedlungsflächen, Straßen, Wege, etc.), und durch ihre räumliche Lage, ihren geometrischen Typ sowie weitere beschreibende Attribute und Beziehungen zu anderen Objekten bestimmt. Jedes Objekt besitzt deutschlandweit eine eindeutige Identifikationsnummer. Das Basis-DLM stellt einen präsentationsneutralen, objektbasierten Vektordatenbestand dar (Bundesamt für Kartographie und Geodäsie 2008). Die Objektarten des Basis-DLM sind im ATKIS® Basis-Objektartenkatalog (AdV 2003) festgelegt.

Die Grundlage des Basis DLM in Baden Württemberg für die geometrische Erfassung der Vektordaten sind Orthophotos im Maßstab 1:10000, die Topographische Karte 1:25000 und die Deutsche Grundkarte 1:5000. Die Lagegenauigkeit beträgt für Punkt- und Linienobjekte ± 3 m. Diese bezieht sich auf die Achsen der Straßen, Schienenbahnen, Gewässer und Leitungen (Landesvermessungsamt Baden-Württemberg 2009a).

3.3.5 Aufbau der Maske

Die Masken werden als Polygon-Shapes vom Projektauftraggeber geliefert. Die Polygon-Shapes werden je Kachel als eigenständiges ESRI-Shapefile geliefert (vgl. Kapitel 3.1.1). An dieser Stelle wird nur eine grobe Skizzierung der Maskengenerierung dargestellt. Die Maskengenerierung wurde in ESRI ArcGIS mit Hilfe des ModelBuilder implementiert.

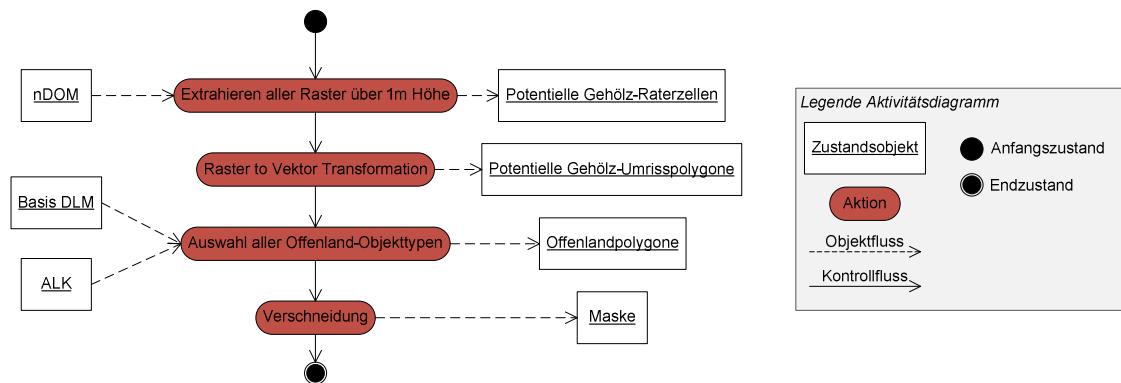


Abbildung 26: Aktivitätsdiagramm der Maskengenerierung

Wie dem Aktivitätsdiagramm der Maskengenerierung (Abbildung 26) zu entnehmen ist, werden zunächst aus dem nDOM alle Rasterzellen mit einem Höhenwert über 1 m extrahiert. Aus diesen Rasterzellen werden mittels einer Vektorisierung Umrisspolygone generiert. Die Umrisspolygone repräsentieren alle Objekte, die höher als 1 m über der Oberfläche liegen. Anschließend werden aus den Quellen Basis-DLM und ALK alle Polygone selektiert und exportiert, die gemäß ihrer Attributierung als Offenland definiert werden können. Die Offenlandpolygone werden abschließend mit den zuvor extrahierten Objekten über 1 m Höhe verschneidet. Die resultierenden Flächen sind diejenigen Flächen (Maske), in denen das Objektmodell „Streuobstbaum“ anwendbar ist.

Abbildung 27 veranschaulicht die Arbeitsschritte der Maskengenerierung anhand von 4 Darstellungen. In Bild A werden die First Pulse Reflexionen zur Verdeutlichung einem Orthophoto überlagert. In Bild B sind die Umrisspolygone nach der Vektorisierung zu sehen. Bild C zeigt die Offenpolygone des entsprechenden Ausschnitts und Bild D das Ergebnis der Verschneidung von Offenpolygone und Umrisspolygonen (invertierte Darstellung der resultierenden Maskepolygone).

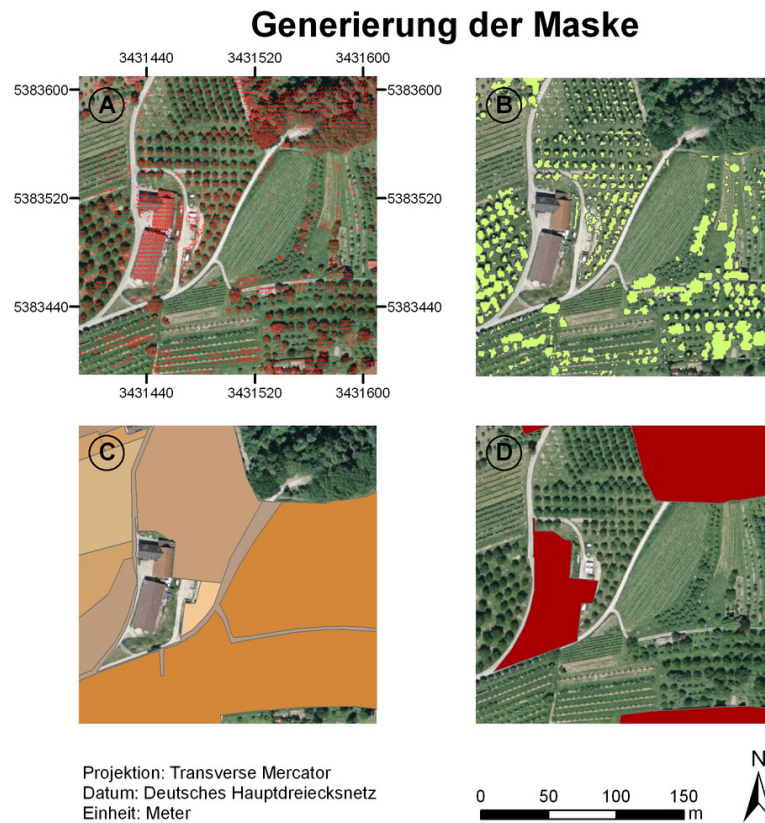


Abbildung 27: Vereinfachte Darstellung der Arbeitsschritte bei der Generierung der Maske

3.3.6 Qualität der Maske

Die räumliche und zeitliche Genauigkeit der Maske ist abhängig von der jeweiligen Genauigkeit der Ausgangsdaten. Fehler in der Maske sind insbesondere auf die zeitliche Verzögerung der Aktualisierung der automatisierten Liegenschaftskarte und des ATKIS Basis DLM zurückzuführen. Die beiden wichtigsten Fehlerklassen (siehe Abbildung 28) sind „nicht aktualisierte Gebäude“ und „ungenau Abgrenzungen topographischer Objekte“ (beispielsweise Wald, Siedlung).

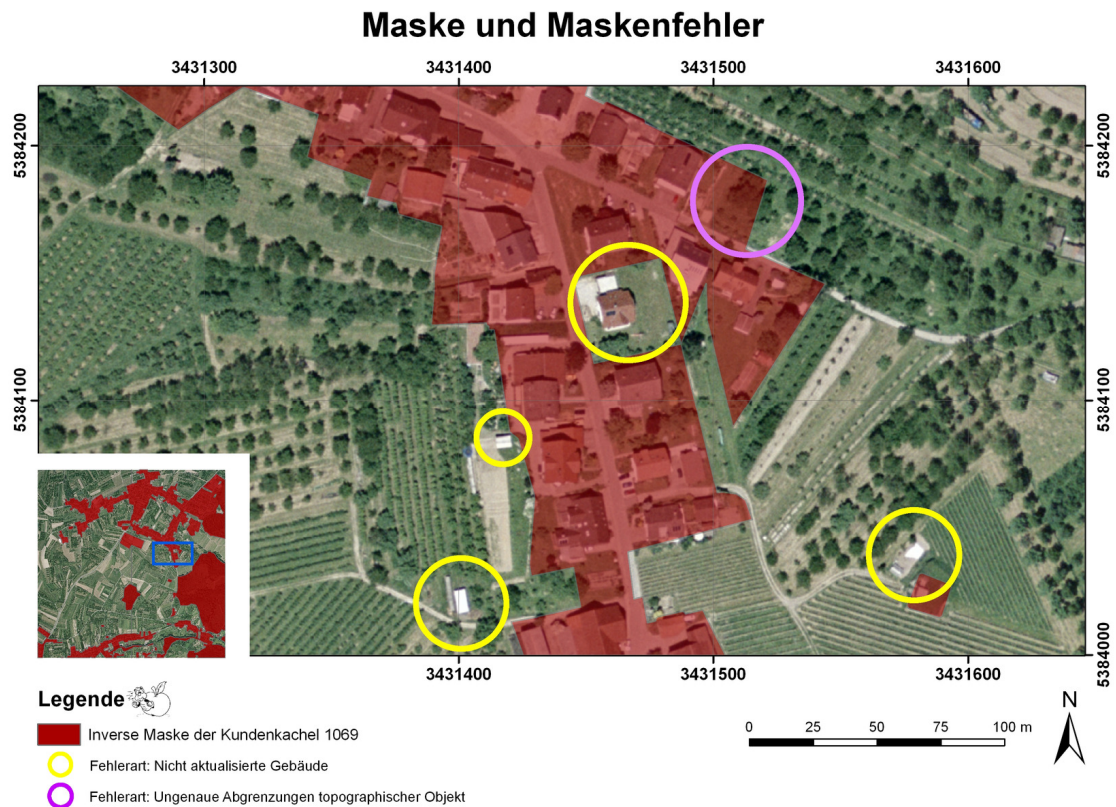


Abbildung 28: Fehler in der generierten Maske

4 Methoden und Werkzeuge

4.1 Methoden der Entwicklung des Algorithmus zur Einzelbaumextraktion

4.1.1 Theorieansatz

Der Theorieansatz des Algorithmus zur Einzelbaumextraktion basiert auf der Entwicklung eines speziell auf die Datengrundlage (ALS-Daten des Landes Baden-Württemberg) abgestimmten Objektmodells im Sinne der objektorientierten Bildanalyse. Grundlage des Objektmodells „Baum“ ist die Synthese von Modellannahmen bestehender Verfahren, die in Kapitel 2.4 zusammengefasst sind. Aus den verschiedenen Arbeiten, insbesondere derjenigen von Straub (Straub 2003) und Bechtel (Bechtel 2007) wurde unter Berücksichtigung der Eigenschaften der vorhandenen ALS-Daten Modellannahmen zusammengetragen.

Modellannahmen des Objektmodells unter Berücksichtigung des Datenmaterials

Die in Tabelle 6 zusammengetragenen Modellannahmen beschreiben das Objektmodell „Baum“, welches im Algorithmus umgesetzt wurde.

Tabelle 6: Objektmodellbeschreibung

Annahme	Beschreibung
A1	Die grobe 3-D Struktur einer Krone ist ein Ellipsoid (siehe Kapitel 2.4)
A2	Eine Krone ist von oben betrachtet ebenfalls elliptisch. Als Spezialfall wird die Krone häufig als kreisförmig betrachtet.
A3	Die 3-D-Hülle einer Krone ist konvex
A4	Die 2-D-Hülle einer Krone ist konvex
A5	Der Baum hat einen höchsten Punkt, die Baumspitze (lokales Maximum)
A6	Die Krone beginnt erst in einer bestimmten Höhe über dem Boden. Diese Höhe kann sich bei verschiedenen Baumarten unterscheiden. Der Umkehrschluss kann ebenfalls angewendet werden: Alles über einer gewissen Höhe ist Teil einer Baumkrone
A7	Eine Baumkrone ist kompakt, d.h. beschränkt und abgeschlossen. Diese Annahme impliziert bei Modellen die als Segmentierungsverfahren Region Growing verwenden, das die Regionen nicht beliebig wachsen dürfen sondern einem Abbruchkriterium genügen müssen
A8	Eine Krone ist einfach zusammenhängend, d.h. frei von Löchern
A9	Der Durchmesser der Krone ist bestimmt. Es gibt einen kleinsten und einen größten Kronendurchmesser
A10	Die Höhe des Stammes ist bestimmt. Es gibt eine kleinste und eine größte Stammhöhe

Weitere Modellannahmen, die auf die vorliegenden ALS-Daten anwendbar sind, jedoch im realisierten Algorithmus zunächst nicht umgesetzt wurden, sind in Tabelle 7 zusammengefasst.

Tabelle 7: Nicht berücksichtigte Modellannahmen

Annahme	Beschreibung
A11	Kronen haben einen bestimmten Abstand voneinander.
A12	Laut Streuobstdefinition darf es nicht mehr als 100 Bäume pro ha geben
A12	<p>Pitkänen et al. definieren in ihrem Modell eine Annahme, nach der der maximal mögliche Kronendurchmesser eines Baumes von dessen Höhe abhängig ist (Pitkänen J. et al. 2004). Die Formel mit den beiden empirisch ermittelten Konstanten a und b lautet:</p> $cw = a + b \cdot h$ <p>cw = Maximaler Durchmesser der Baumkrone h = Baumhöhe $a = 1,2$ $b = 0,16$</p> <p>Formel 9: Abhängigkeit des maximalen Kronendurchmessers von der Baumhöhe</p>

Segmentierung und Merkmalsextraktion

Zentraler Bestandteil der im Algorithmus umgesetzten Einzelbaumextraktion ist die Segmentierung mit Hilfe der Watershed-Transformation. Bei der anschließenden Merkmalsextraktion wurden sowohl Merkmale der Segmente als auch Merkmale der innerhalb der Segmente liegenden vektoriiellen Ausgangsdaten (ALS-Daten) bestimmt.

Tabelle 8: Objektmerkmale

Merkmal	Beschreibung
M1	Fläche der konvexe Hülle
M2	Anzahl der First Pulse Reflexionspunkte innerhalb der konvexen Hülle
M3	Anzahl der Last Pulse Reflexionspunkte innerhalb der konvexen Hülle
M4	Kompaktheit
M5	Kreisförmigkeit
M6	Mittlere Höhe der normalisierten First Pulse Reflexionspunkte
M7	Maximale Höhe der normalisierten First Pulse Reflexionspunkte
M8	Stammpunkt (Point-Feature))
M9	Verhältnis von Höhe zu Fläche
M10	Anzahl Veg-Last / (Veg.-Last + Veg.-First); Maß im Wertebereich [0,1]

4.1.2 Vorgehensweise bei der Entwicklung des Algorithmus

Softwareentwicklungsmethode

Die Vorgehensweise der Entwicklung des Einzelbaumextraktionsalgorithmus lehnt sich an das Vorgehensmodell des Extreme Programming (XP) aus der objektorientierten Softwareentwicklung an (Bunse & von Knethen 2008). Dieses Vorgehensmodell beruht auf dem Prinzip der frühen und intensiven Einbeziehung aller Beteiligten und besteht aus zwei Phasen: einer Planungsphase zur Identifizierung und Festlegung der Anforderungen und einer iterativen Entwicklungsphase zur Realisierung des Softwaresystems.

Anforderungen und Rahmenbedingungen

In Zusammenarbeit mit dem KMU aus dem Bereich Fernerkundung wurden folgende grundlegende Anforderungen und Rahmenbedingungen in Zusammenhang mit der Realisierung des Algorithmus zur Einzelbaumextraktion festgelegt:

- Der Prozess muss batchfähig sein, d.h. die komplette Verarbeitung muss im Hintergrund laufen
- Der Prozess muss flexibel genug aufgebaut werden, um auf vergleichbare Fragestellungen transformiert werden zu können
- Der Prozess muss flexibel auf die IT-Infrastruktur eines mittelständischen Unternehmens anpassbar sein. Dies bedeutet, dass der Prozess weitgehend unabhängig von einer konkreten Systemplattform entwickelt werden muss. Mögliche Zielplattformen sind Windows und Linux. Weiterhin sollen nur solche Systemkomponenten verwendet werden, die mit einem absehbaren Aufwand durch alternative Komponenten ausgetauscht werden können.
- Die erste Version des Algorithmus wird im Sinne einer schnellen und effizienten Implementierung unter Verwendung von Matlab realisiert. Bei der Entwicklung des Algorithmus ist darauf zu achten, dass der funktionale Ablauf möglichst auf andere Entwicklungssysteme übertragbar ist.

Ablauf der Softwareentwicklung

In einer Rapid-Prototyping-Phase wurden unterschiedliche, in der bestehenden Literatur beschriebene Verfahren erprobt. Dabei hat sich herausgestellt, dass die Segmentierung den für die Ergebnisse wesentlichen Schritt des Algorithmus darstellt. Im Einzelnen wurden folgende Ansätze verifiziert, die sich jedoch bei der Anwendung auf die vorliegenden Daten als nicht geeignet erwiesen haben:

- Verschiedene Region-Growing-Varianten auf der Grundlage morphologischer Operatoren
 - Bestimmen der Saatpunkte mittels unterschiedlicher morphologischer Operatoren

- Region-Growing innerhalb der vektoriellen 3-D-Laserscan-Daten auf der Grundlage einer Delaunay-Triangulation
- Multiskalare Segmentierung ebenfalls auf Basis morphologischer Operatoren auf drei verschiedenen Skalenebenen zur Extraktion von großen, mittleren und kleinen Bäumen.

4.1.3 Werkzeuge der Entwicklung

Als Entwicklungswerkzeug wurde Matlab (Version 7.8) mit den Erweiterungen Image Processing Toolbox, Mapping Toolbox und der Statistics Toolbox verwendet. Die vom Algorithmus generierten Ergebnisse wurden mit Hilfe von Funktionen aus der Mapping Toolbox als ESRI-Shapefiles gespeichert. Diese Ergebnisse konnten anschließend in ArcGIS (2-D) oder ArcScene (3-D) visuell überprüft werden. Dazu wurden die Ergebnisse den für die jeweiligen Testflächen vorliegenden Orthophotos, sowie den Ausgangsdaten (ALS-Daten und Masken) überlagert.

Die für die Entwicklung notwendigen Testdaten (ALS-Daten und Masken) wurden mittels eigener Python-Skripte, unter Verwendung der Bibliothek „GDAL“ (*Geospatial Data Abstraction Library*) zum Lesen und Schreiben von ESRI-Shapefiles, aus den Eingangsdaten generiert.

4.1.4 Validierung des Einzelbaumextraktionsalgorithmus

Die Validierung des Einzelbaumextraktionsalgorithmus erfolgt auf Grundlage der in Kapitel 2.3.6 beschriebenen Qualitätsmaße **Vollständigkeit** und **Korrektheit**. Es wurden verschiedene Referenzflächen ausgewählt (siehe Kapitel 3.1.3). Anhand der Orthophotos wurden von einem unabhängigen Experten in insgesamt 6 Flächen von je 200 m x 200 m alle erkennbaren Einzelbäume gemäß folgender unscharfer Vorschrift in zwei Referenzdatensätze digitalisiert: Alle Bäume, deren Kronenfläche kleiner als etwa 2 m² sind (entspricht bei einer Auflösung der Orthophotos von 25 cm einer Fläche von 32 Pixeln), werden im Referenzdatensatz „kleine Bäume“, alle übrigen Bäume im Referenzdatensatz „große Bäume“ eingepflegt.

4.2 Methoden der Entwicklung des Massenprozessierungssystems

4.2.1 Methodische Vorgehensweise bei der Recherche

Ziel des Massenprozessierungssystems ist der Aufbau eines wiederverwendbaren Systems zur automatischen Prozessierung umfangreicher fernerkundlicher Datenbestände. Zunächst wurden in Zusammenarbeit mit dem KMU Anforderungen

und Rahmenbedingungen an ein Massenprozessierungssystem erarbeitet, die im Folgenden zusammengefasst sind:

- Das System muss unter Linux betrieben werden
- Das System muss kostenfrei sein und eine Anpassung an die vorliegende Systemumgebung beim Projektpartner muss möglich sein
- Das System muss verschiedenartige Jobs sowohl unter Windows, als auch unter Linux verteilen und steuern können. Es muss also über eine Steuerskomponente für unterschiedliche Knotenplattformen verfügen.
- Das System muss über eine Monitoring-Komponente verfügen, mit der es möglich ist, dass auch Nicht-Fachleuten den aktuellen Systemstatus beobachten und bewerten können.

Anhand dieser Anforderungen wurde eine Recherche nach Systemen durchgeführt. Schlüsselwörter der Recherche waren die Begriffe: Job Queuing Systeme, Batchsysteme, Grid-Middleware, Clustersysteme, Workflow-Managementsysteme, Continuous Integration Systems (*Begriff aus der Software-Entwicklung, der den Prozess des regelmäßigen, vollständigen Neubildens und Testens von Anwendungen beschreibt*). Die Recherche konzentrierte sich auf eine Literaturrecherche zu den verschiedenen, oben aufgeführten Schlüsselwörtern, sowie eine Internetrecherche bei Portalen entsprechender Anbieter und offener Entwicklungsprojekte. Im Anschluss an die Recherche wurde eine Testumgebung für zwei potentielle Systeme eingerichtet. Im Rahmen der Testumgebung wurde die Prozessverteilung und Steuerung anhand einfacher Batch-Skripte untersucht:

- Hudson (<https://hudson.dev.java.net>)
- BOINC (<http://boinc.berkeley.edu>)

4.2.2 Katalogisierung der Eingangsdaten

Beim Übergang des Algorithmus von der Verarbeitung einzelner Datensätze hin zur Verarbeitung von Massendaten müssen im Vorfeld zwei zentrale Fragen geklärt werden. Wie sind die Massendaten organisiert und wie soll die Verarbeitung dieser Daten ablaufen?

Da die Eingangsdaten oftmals nicht den Erfordernissen einer Massenverarbeitung genügen, müssen diese in einem Vorverarbeitungsschritt entsprechend angepasst werden. Im vorliegenden Fall werden die Eingangsdaten im Rahmen einer Vorverarbeitung reorganisiert. Damit später die Ausgangsdaten der Prozessierung den Eingangsdaten zugeordnet werden können, wird ein Katalog in Form einer Datenbank angelegt. In der Datenbank werden sowohl alle Eingangsdaten als auch alle Prozessdaten mit deren Metainformationen (Ablageort) hinterlegt. Im konkreten Fall

werden alle Dateinamen der Eingangsdaten einschließlich der vollständigen Pfadangabe in der Datenbank gespeichert. Ebenso werden alle Informationen der Prozessdaten gespeichert. Eine Fremdschlüsselbeziehung verbindet die Prozessdaten mit den zugehörigen Eingangsdaten.

4.2.3 Aufbau des Massenprozessierungssystems

Konzeption und Aufbau des Massenprozessierungssystems ist Teil des Ergebnisses und werden in Kapitel 6 ausführlich beschrieben.

4.2.4 Bewertung des Massenprozessierungssystems

Die Bewertung des Massenprozessierungssystems basiert einzig auf der Geschwindigkeit, mit der die Daten verarbeitet werden können. Um dieses Geschwindigkeit zu bewerten, wurde ein Vergleichssystem auf der Grundlage eines leistungsfähigen Rechners (*Intel(R) Core(TM) i7 CPU 9200 2.67 GHz, 8 Prozessoren*) aufgebaut. Der Rechner verfügt über acht Prozessoren, die denjenigen der Desktopclients (*Dual-Core-Rechner, 2.8 GHz, 2 Prozessoren*) aus dem BOINC-System entsprechen. Sowohl der Vergleichsrechner als auch die Desktopclients des BOINC-Systems werden unter Windows XP betrieben.

Zur Bewertung werden die Laufzeiten der Verarbeitung von je 500 Eingangskacheln in jedem der beiden Systeme aufgezeichnet. Als Bewertungsmaß wird der Mittelwert der Laufzeit je Systemumgebung herangezogen und diskutiert.

5 Ergebnis des Algorithmus zur Einzelbaumextraktion

5.1 Algorithmus zur Einzelbaumextraktion

5.1.1 Ein- und Ausgangsdaten des Algorithmus

Der Algorithmus der Einzelbaumerkennung erhält als Eingangsdaten die Maske und die vier klassifizierten ALS-Datensätze in Form von Punktwolken. Als Ergebnis liefert der Algorithmus die Positionen und Konturen der gefundenen Objekte in Form von ESRI-Shapefiles. Abbildung 29 fasst die Schnittstellen des Algorithmus in Form eines Aktivitätsdiagramms zusammen. Im weiteren Verlauf dieses Kapitels wird detailliert auf die Einzelschritte des Algorithmus eingegangen.

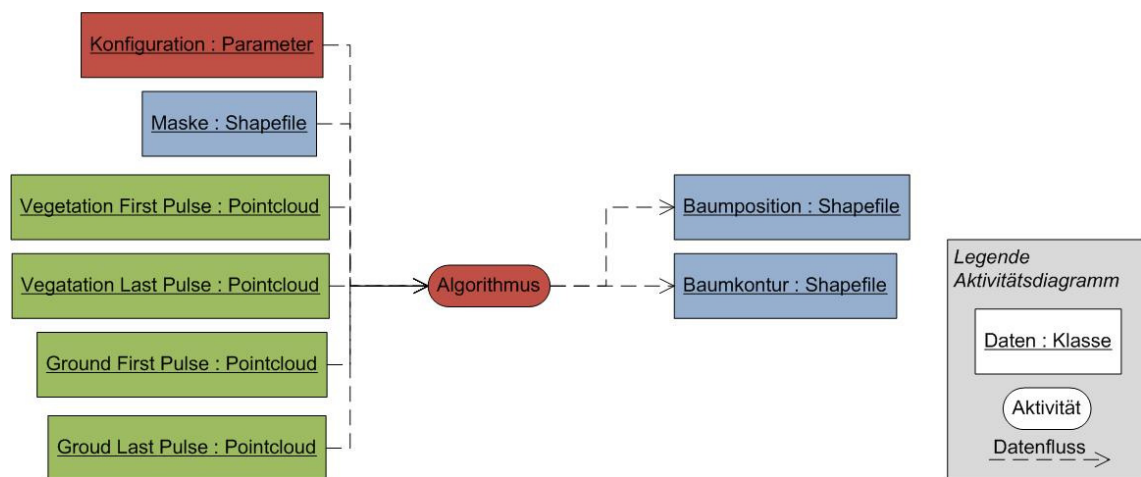


Abbildung 29: Eingangs- und Ausgangsdaten des Algorithmus

Die Erläuterung der Einzelschritte des Algorithmus wird beispielhaft anhand eines Testdatensatzes aus dem Untersuchungsgebiet veranschaulicht. Testdatensatz ist ein Ausschnitt von 220 m x 220 m aus der Kachel 1070. Der Ausschnitt entspricht der Prozesskachel 336009 (siehe Kapitel 4.2.2). In der Abbildung sind zusätzliche die reorganisierten First Pulse Boden- und Vegetationsdaten zu sehen.

Prozesskachel 336009 mit Überlappungsbereich (10 m)

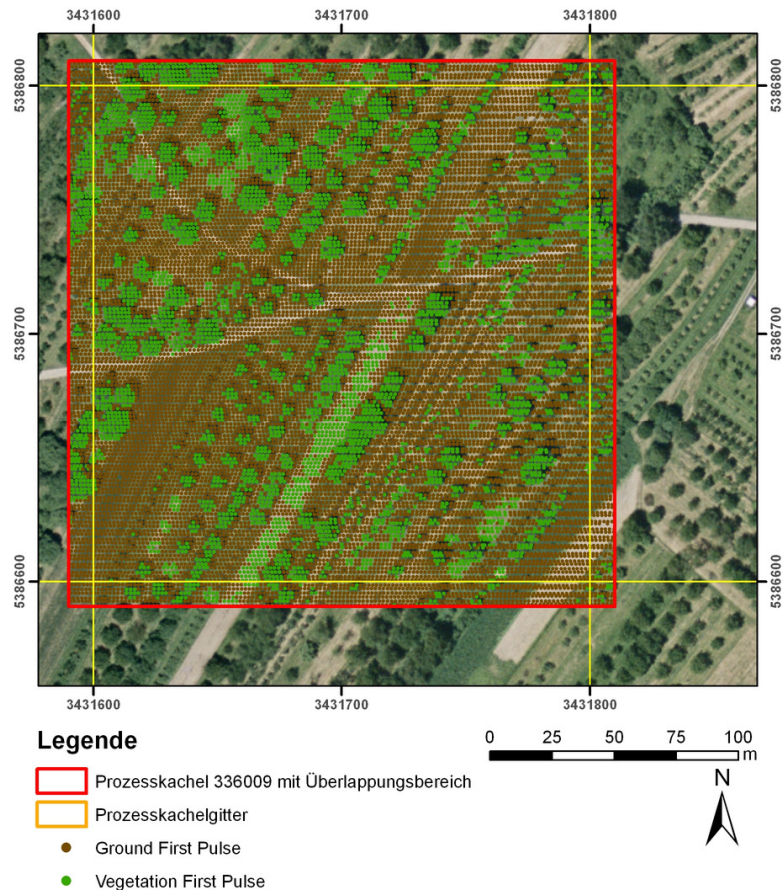


Abbildung 30: Darstellung einer Prozesskachel mit First Pulse-Daten

5.1.2 Konfigurationsparameter des Algorithmus

Alle Konstanten und Konfigurationsparameter, die das Verhalten und das Ergebnis des Algorithmus beeinflussen, werden aus dem Prozess ausgelagert und als Übergabeparameter deklariert. Somit kann die Konfiguration des Algorithmus, sowie die Auswahl der Ein- und Ausgangsdaten von der Logik des Algorithmus getrennt werden. Die Konfigurationsparameter des Algorithmus können in drei Gruppen unterteilt werden:

Parameter der Ein- und Ausgangsdaten

VEF_FILE	=	'\project\orchard\processdata\33\60\09\input\vef.txt'
VEG_FILE	=	'\project\orchard\processdata\33\60\09\input\veg.txt'
GRF_FILE	=	'\project\orchard\processdata\33\60\09\input\grf.txt'
GRD_FILE	=	'\project\orchard\processdata\33\60\09\input\grd.txt'
MASK_FILE	=	'\project\orchard\processdata\33\60\09\input\mask.shp'
OUTPUT_PATH	=	'\project\orchard\processdata\33\60\09\output'

Parameter des Untersuchungsgebietes

BOUNDING_LEFT	= 3.380.000	# Gauß-Krüger, Projektkonstante
BOUNDING_TOP	= 5.265.000	# Gauß-Krüger, Projektkonstante
NUM_OF_PROCESS_TILE_COLS	= 1300	# Projektkonstante
NUM_OF_PROCESS_TILE_ROWS	= 1300	# Projektkonstante

Parameter der Prozesskachel-Verarbeitung

PADDING_SIZE	= 10	# Ueberlappungsbereich, [Meter]
RASTER_SIZE	= 1.0	# Generierung der oberflaechenmodelle, [Meter]
MIN_TREE_HEIGHT	= 1.2	# Mindestbaumhoehe aus Baummodell, [Meter]

5.1.3 Ablauf des Algorithmus

Zunächst werden die Einzelschritte des Algorithmus in Abbildung 31 in Form eines Aktivitätsdiagramms dargestellt. Anschließend werden die Einzelschritte im Detail anhand des Beispieldatensatzes erläutert.

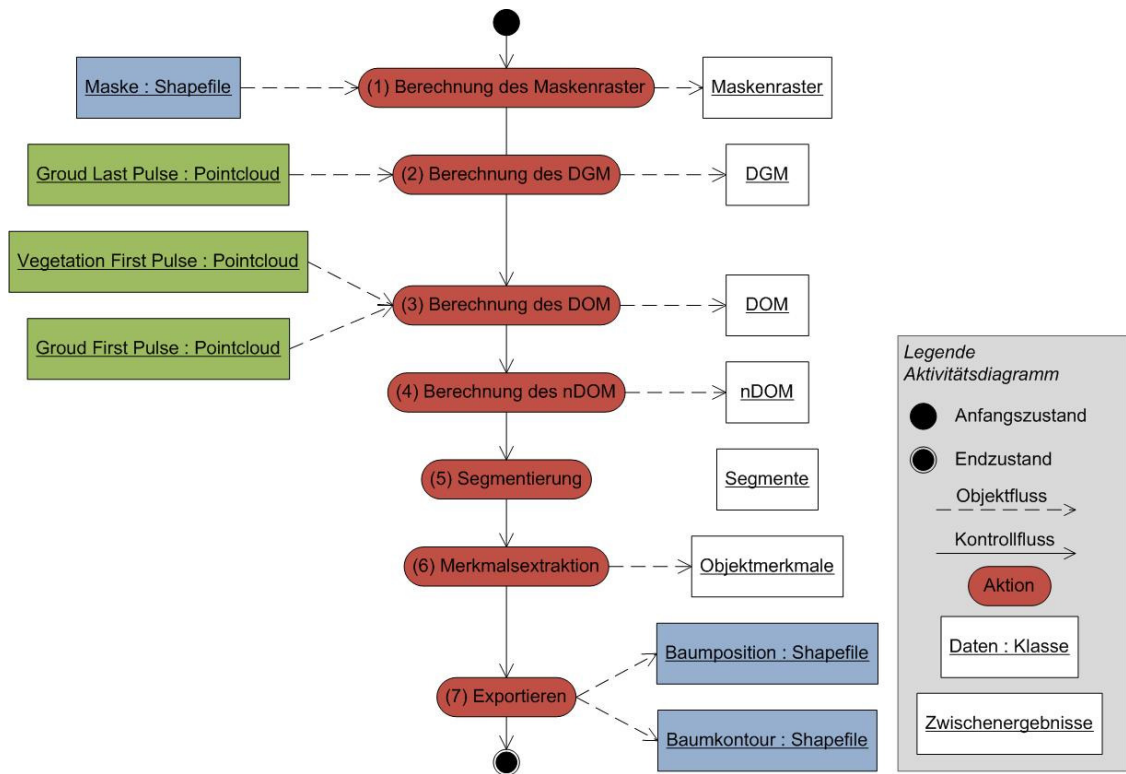


Abbildung 31: Detailliertes Aktivitätsdiagramm mit alle Teilschritten

Schritt 1: Berechnung des Maskenraster

Im ersten Schritt des Algorithmus werden zunächst die Maskenpolygone, die in Form eines ESRI-Shapefiles vorliegen, in ein Raster konvertiert. Das resultierende Maskenraster enthält anschließend für alle relevanten Zellen den Wert „1“, für alle Zellen, in denen nicht nach Bäumen gesucht werden soll, den Wert „0“. Das

Maskenraster wird in Schritt 4 mit dem nDOM multipliziert und ergibt das gefilterte nDOM, welches anschließend in die Segmentierung übergeben wird.

Schritt 2: Berechnen des DGM

Kernstück des Algorithmus ist der Schritt 5, die Segmentierung des normalisierten Oberflächenmodells auf der Basis morphologischer Operatoren. Zu diesem Zweck müssen die vektoriellen Ausgangsdaten zunächst in Rasterdaten konvertiert werden.

In Schritt 1 wird aus den Vektordaten der Last Pulse Bodenpunkte das DGM errechnet. Bei dieser Operation wird auf die Berücksichtigung der Last Pulse Vegetationspunkte verzichtet, da sich im Rahmen von Tests herausgestellt hat, dass die die Last Pulse Vegetationspunkte insbesondere in Bereichen von Vegetation fehlerhaft klassifiziert sind und das Ergebnis der Segmentierung maßgeblich verschlechtern.

In der Regel werden ALS-Daten mittels eines Samplings in Rasterdaten konvertiert. Dabei muss jedoch darauf geachtet werden, dass das Abtasttheorem eingehalten wird. Bei einer mittleren Punktdichte von 1,2 Punkte pro m² müsste im vorliegenden Fall also mit einer minimalen Rasterweite von 2,5 m gearbeitet werden. Andererseits führen die morphologischen Operatoren, die bei der Segmentierung zum Einsatz kommen, zu weiteren Informationsverlusten um einen Faktor > 2. Die kleinstmögliche Objektgröße würde in diesem Falle über 5 m betragen.

Aufgrund der Objektgröße von Streuobstbäumen wurde eine alternative, auf die Daten abgestimmte, Rasterdatengenerierung realisiert. Dabei werden die vektoriellen ALS-Daten zunächst auf ein Raster mit der Rasterweite 1 m gesampelt. Dabei bleiben jedoch viele Rasterzellen nicht berücksichtigt. Abbildung 32 veranschaulicht dieser Rasterung anhand eines Ausschnitts mit First Pulse Bodenpunkten (braun) und First Pulse Vegetationspunkten (grün).

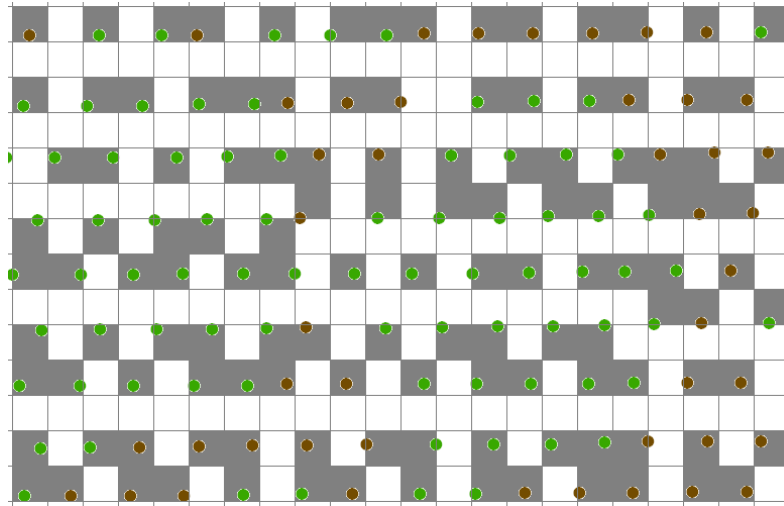


Abbildung 32: Übetragung der First Pulse Daten auf ein Raster mit der Rasterweite 1 m

Die nicht gefüllten Rasterzellen werden anschließend mit Hilfe einer vereinfachten Interpolation mit dem Mittelwert der signifikanten Nachbarzellen (Nachbarzellen, die einen Wert enthalten) gefüllt. Dabei wächst der Suchradius iterativ solange, bis wenigstens zwei signifikante Nachbarzellen gefunden sind. Abbildung 33 veranschaulicht anhand zweier Screenshots aus der Matlab-Entwicklungsumgebung das Schließen der Lücken. Im linken Bild sind die durch Vegetation bedingten Lücken und die First Pulse Bodenpunkte deutlich zu erkennen; Das Bild ist auf eine Höhe von 160 m normiert, damit die topografische Struktur annähernd erkennbar ist. Die Streifenbildung im Ergebnis (rechts) rührt nicht von etwaigen Richtungsabhängigkeiten des Algorithmus sondern ist topografisch bedingt (Phänomen Wölbäcker).

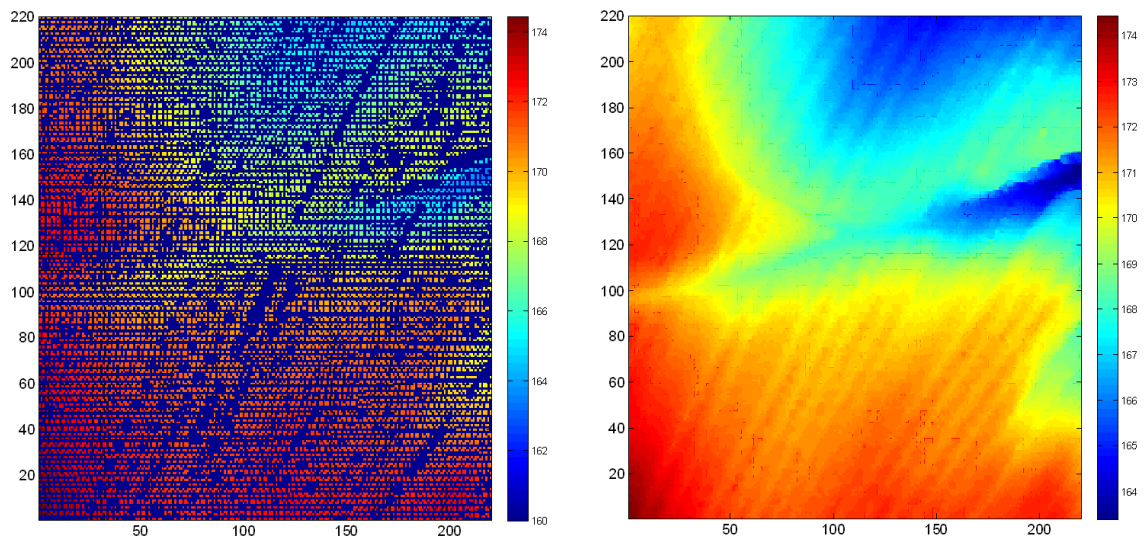


Abbildung 33: Gesampelte Last Pulse Bodenpunkte mit Lücken vor der Interpolation (links) und danach (rechts)

Schritt 3: Berechnung des DOM

Im zweiten Schritt wird aus der Kombination der beiden First Pulse Datensätze (First Pulse Bodenpunkte und First Pulse Vegetationspunkte) ebenfalls ein Raster generiert. Dabei wird derselbe Algorithmus wie zur Berechnung des DGM verwendet. Abbildung 34 stellt die gesampelten Pulse-Daten (links, normiert auf Null = 160 m) sowie das Resultat der Interpolation gegenüber.

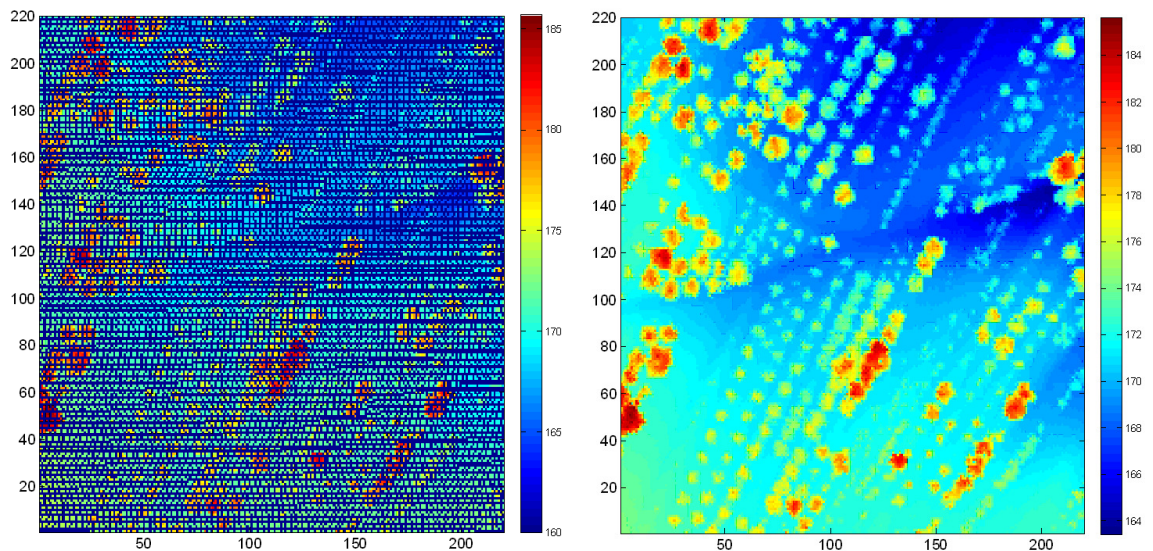


Abbildung 34: Gesampelte First Pulse Daten mit Lücken vor der Interpolation (links) und danach (rechts)

Schritt 4: Berechnung des nDOM

Das normalisierte Oberflächenmodell resultiert aus der Differenz von DOM und DGM. Das Ergebnis wird mit dem Maskenraster multipliziert. Daraus ergibt sich ein gefiltertes nDOM in dem nur noch die nicht maskierten Zellen einen Wert ungleich „0“ haben. Anschließend werden alle Zellen mit einem Wert kleiner der minimalen Objekthöhe (MIN_TREE_HEIGHT) ebenfalls auf „0“ gesetzt.

Das vorliegende Anschauungsbeispiel enthält keine ausmaskierten Bereiche, da es in der vorliegenden Algorithmusbeschreibung zentral um den Aspekt der Baumextraktion geht.

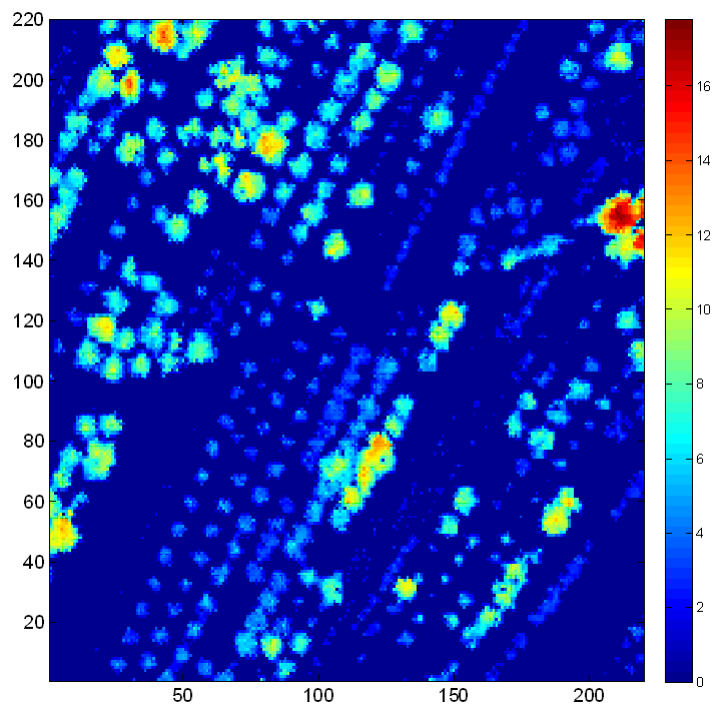


Abbildung 35: Maskiertes und höhengefiltertes nDOM

Das resultierende, maskierte und höhengefilterte nDOM ist Ausgangsprodukt der Segmentierung. Die Baumstrukturen sind in Abbildung 35 bereits deutlich erkennbar.

Schritt 5: Segmentierung

Grundlage der Segmentierung ist die Watershed-Transformation (siehe Kapitel 2.3.3). Diese bedarf unter Verwendung der entsprechenden Matlab-Funktion einer vorherigen Aufbereitung des zuvor errechneten nDOM. Zunächst wird das nDOM in ein Binärbild konvertiert und dieses wiederum invertiert. Anschließend wird mit Hilfe der morphologischen Operation „Distanztransformation“ ein Distanzbild dieses inversen, binären nDOMs berechnet. Die Distanztransformation errechnet für jede Zelle des Ausgangsbildes mit dem Wert „0“ den Abstand zwischen dieser Zelle und der nächst gelegenen Zelle mit einem Wert ungleich „0“. Abbildung 36 veranschaulicht diese vorbereitenden Schritte anhand des Beispieldatensatzes.

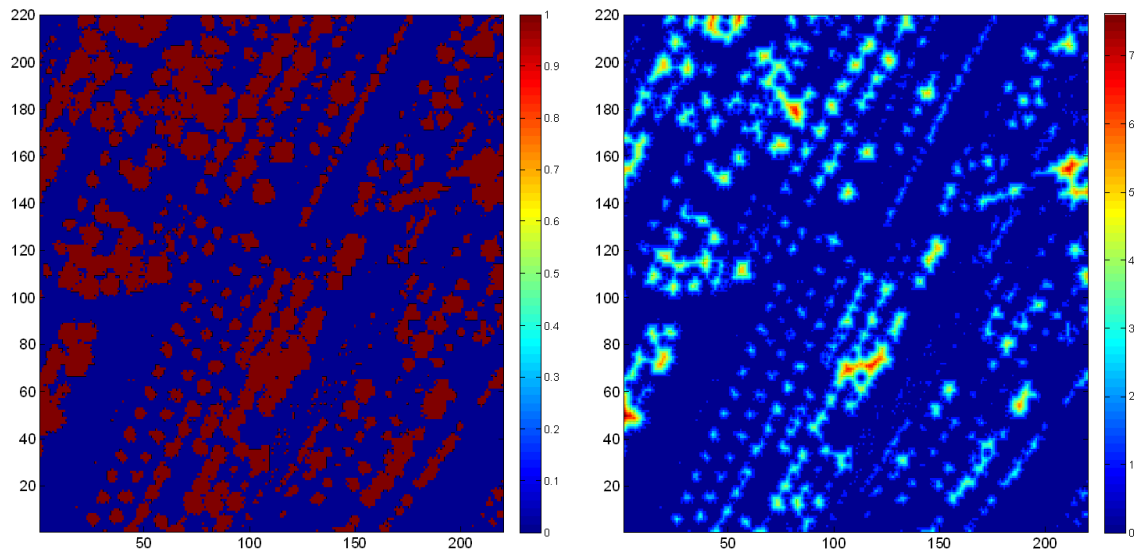


Abbildung 36: Binäres nDOM und Distanzbild des invertierten binären nDOM

Bevor nun die Watershed-Transformation auf das Bild angewendet werden kann, muss das Distanzbild negiert werden, so dass die „Berge“ zu „Tälern“ werden (siehe Kapitel 2.3.3). Abbildung 37 stellt das negierte Distanzbild (links) sowie das Ergebnis der Watershed-Transformation dar. Das Ergebnisbild der Watershed-Transformation besteht aus Segmenten und Trennlinien. Die Trennlinien haben den Zellenwert „0“. Die Zellen eines Segments besitzen alle den gleichen numerischen Wert. Jedes Segment hat einen eindeutigen Wert und kann somit durch logische Bildoperatoren extrahiert werden. Aus der Farbpalette des Ergebnisbildes kann abgelesen werden, dass die Watershed-Transformation mehr als 500 Segmente ermittelt hat.

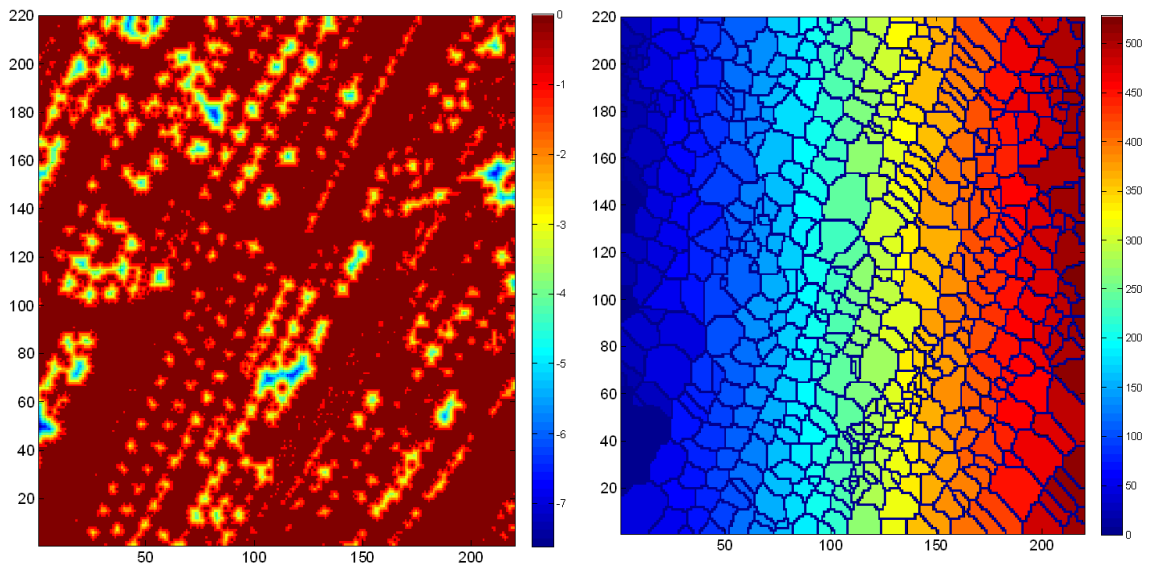


Abbildung 37: Negiertes Distanzbild und Ergebnis der Watershed-Transformation

Abbildung 38 veranschaulicht anhand einer Falschfarbendarstellung das Ergebnis der Segmentierung. Die arithmetische Kombination von Watershed-Transformation und

Distanzbild lässt erkennen, dass die zu suchenden Objekte durch die Transformation getrennt werden.

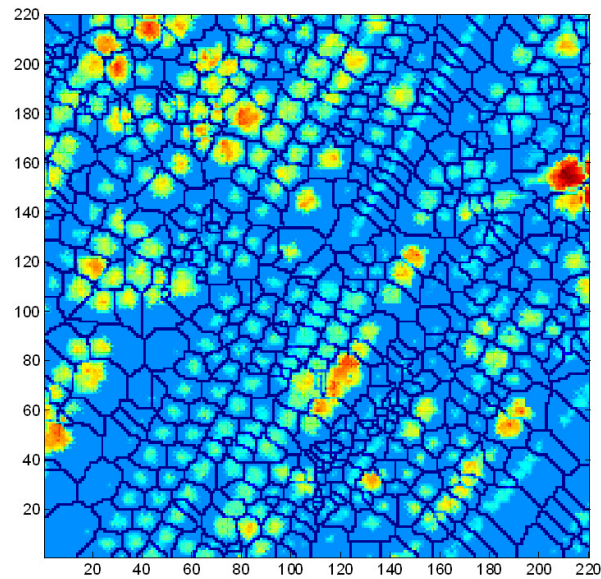


Abbildung 38: Falschfarbendarstellung der Watershed-Transformation

Schritt 6: Merkmalsextraktion

Das Ergebnis der Watershed-Transformation ist Grundlage der Merkmalsextraktion. Bei der Merkmalsextraktion werden in einer Schleife alle Segmente einzeln untersucht. Im Folgenden wird der Ablauf je Segment beschrieben.

Zunächst wird ein binäres Bild erzeugt, in dem nur die Zellen des zu untersuchenden Segments den Wert „1“ erhalten. Diese Segmentmaske wird mit dem aus Schritt 5 bekannten binären nDOM multipliziert. Das Ergebnis ist ein binäres Bild, in dem nur diejenigen Zellen eines Segments ungleich „0“ sind, welche die Objektmindesthöhe (MIN_TREE_HEIGHT) überschreiten. Auf Grundlage der daraus resultierenden Baummaske werden mit Hilfe verschiedener Operationen aus der Matlab-Image Processing Toolbox die zu extrahierenden Merkmale bestimmt. Mit Hilfe der Operation „regionprops“ können verschiedene Merkmale der Baummaske bestimmt werden. Grundlage der meisten Merkmale ist die, ebenfalls mit dieser Operation errechnete, konvexe Hülle der Baummaske.

- **Fläche:** Fläche der konvexen Hülle
- **Exzentrizität:** Maß für die Abweichung einer Ellipse, welche die konvexe Hülle umschließt, von einer flächengleichen Kreisform
- **Äquivalenter Durchmesser:** Durchmesser eines Kreises, der die gleiche Fläche hat wie die Baummaske
- **Ausdehnung:** Verhältnis zwischen der Anzahl Pixel in der Region „Baummaske“ und dem umhüllenden, rechteckigen Rahmen (bounding box)

- **Hauptachsenlänge:** Länge in Pixel der Hauptachse einer Ellipse, welche die gleiche Fläche wie die konvexe Hülle hat und diese minimal umschließt
- **Nebenachsenlänge:** Länge in Pixel der Nebenachse
- **Solidität:** Verhältnis der Anzahl Pixel innerhalb der konvexen Hülle und der Anzahl Pixel innerhalb der Region.

Anschließend werden die vektoriellen Ausgangsdaten (ALS-Daten) untersucht, die innerhalb der konvexen Hülle des Rasters liegen. Es werden also anhand der errechneten Bildmaske Informationen aus den Ursprungsdaten extrahiert. Voraussetzung hierfür ist die Berechnung des Voronoi-Diagramms aller First Pulse Punkte mit Hilfe der Delaunay Triangulation (siehe Kapitel 2.3.1).

Es werden folgende zusätzliche Merkmale ermittelt:

- **Erstreflexionen:** Anzahl der First Pulse Punkte (Bodenpunkte und Vegetationspunkte) innerhalb der konvexen Hülle der Baummaske, die über der geforderten Objektmindesthöhe liegen.
- **Gesamtreflexionen:** Anzahl aller ALS-Datenpunkte innerhalb der konvexen Hülle der Baummaske
- **Baumposition:** Der mit der Höhe gewichteter Mittelwert aller Erstreflexionen
- **Maximale Baumhöhe:** Höchste Reflexion aller Erstreflexionen
- **Mittlere Baumhöhe:** Mittelwert der Erstreflexionen

Schritt 7: Exportieren

Die Ergebnisse werden mit der Funktion „*shapewrite*“ aus der Matlab-Mapping Toolbox als ESRI-Shapefile gespeichert. Es wird ein Point-Feature für die Baumpositionen und ein Polygon-Feature für die Baumkonturen gespeichert. Zur späteren Auswertung und Analyse der Daten wird je Objekt zusätzlich eine je Kachel eindeutige Objektnummer als Attribut hinterlegt.

5.2 Validierung der Ergebnisse

5.2.1 Performance

Es wurden verschiedene Untersuchungen der Performance durchgeführt. Kritisch ist die Größe der Prozesskacheln im Zusammenhang mit den Überlappungsbereichen. Werden die Prozesskacheln zu groß gewählt, sinkt die Performance des Algorithmus deutlich. Bei Prozesskacheln der Größe 400 m x 400 m, also etwa einer Vervierfachung der vorherigen Prozesskachelfläche von 200 m x 200 m, steigt die Berechnungszeit um etwa das Achtfache. Ursache sind Teilschritte innerhalb des Algorithmus, die nicht linear mit der Datenmenge steigen, wie beispielsweise die Berechnung der Delaunay-Triangulation (siehe Kapitel 2.3.1).

5.2.2 Genauigkeitsanalyse

Die Genauigkeitsanalyse basiert auf den erstellten Referenzdaten sowie den Ergebnisse der Einzelbaumextraktion in zuvor definierten Referenzflächen (vergleiche Kapitel 4.1.4). Abbildung 39 stellt die Referenzdaten den Ergebnissen der Einzelbaumextraktion gegenüber.

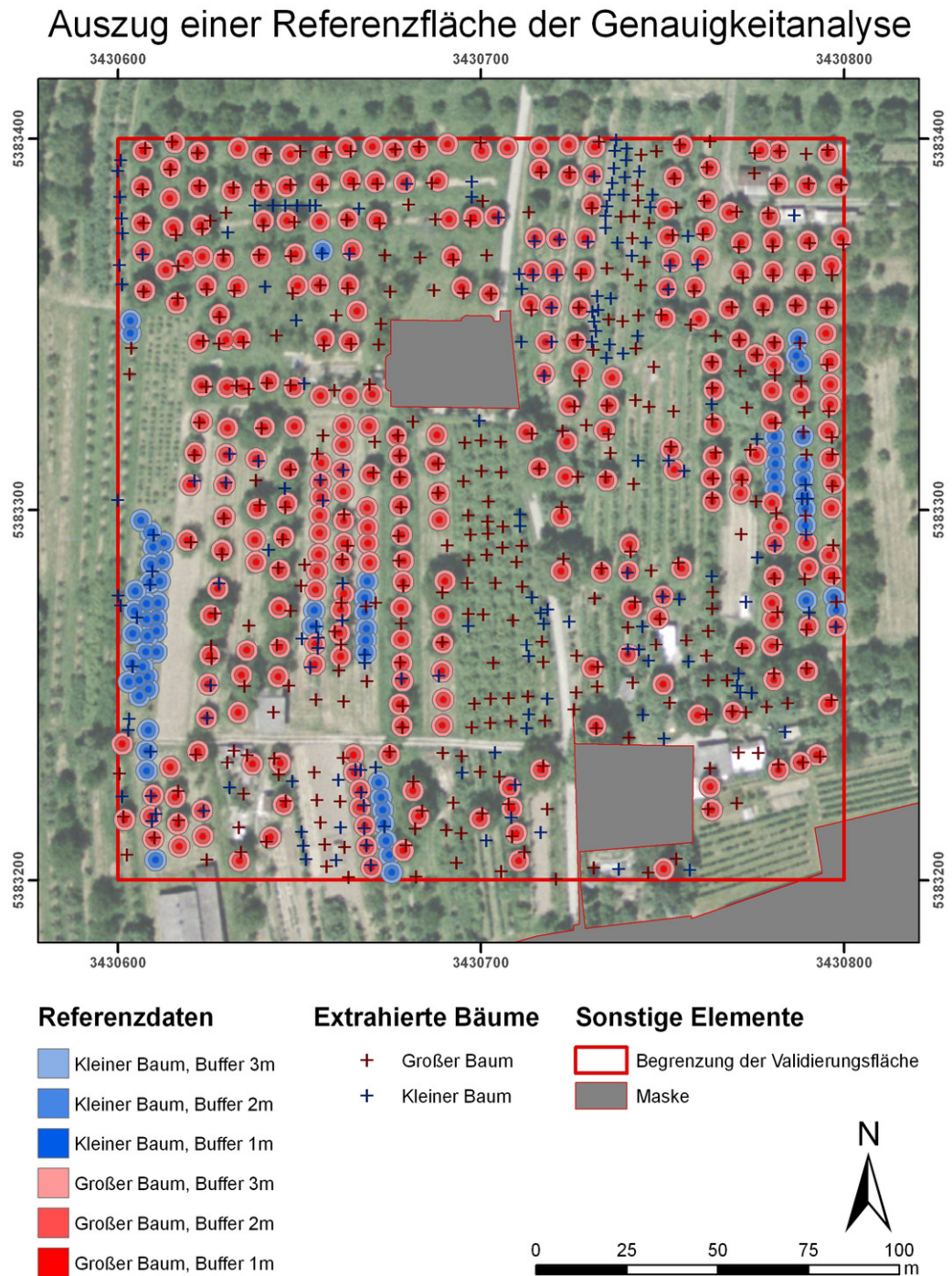


Abbildung 39: Auszug aus den Referenzflächen mit gepufferten Validierungsdaten und den Bäumen der Objektextraktion

Aufgrund des großen Zeitabstands zwischen Laserscan-Befliegung und der Aufnahme der Orthophotos besteht teilweise ein signifikanter Unterschied zwischen den Referenzdaten und den Extraktionsergebnissen. So sind auf den Orthophotos Neubestände erkennbar, die in den ALS-Daten vollständig fehlen. Andererseits enthält das Ergebnis der Einzelbaumextraktion Bäume, die auf den 5 Jahre später erstellten Orthophotos nicht mehr vorhanden sind. Aus diesem Grund wird eine getrennte Validierung für kleine (Fläche kleiner 2 m^2) und große Bäume durchgeführt. Bei der Erstellung der Referenzdaten wurde diese Trennung aufgrund einer visuellen Abschätzung der Baumkronenfläche durchgeführt. Die Extraktionsergebnisse wurden auf der Basis der Fläche der konvexen Hülle aller Firstreflexionen innerhalb des Objektes in die beiden Gruppen „groß“ und „kleiner“ unterteilt.

Als Toleranzmaß fließt der Fehlerabstand in Form eines Buffering der Referenzpunkte in die Validierung ein. Es werden drei getrennte Validierungen mit den Fehlerradien 1 m, 2 m und 3 m durchgeführt (vergleiche Buffering in Abbildung 39).

Die Ergebnisse der Validierung sind in Tabelle 9 und Tabelle 10 zusammengefasst.

Tabelle 9: Vollständigkeit und Korrektheit der Objektklasse „kleine Bäume“

Kleine Bäume (Kronenfläche $< 2 \text{ m}^2$)					
Fehler-radius	FP	FN	TP	Vollständigkeit (Producer's Accuracy)	Korrektheit (User's Accuracy)
1 m	984	627	14	2,2 %	1,4 %
2 m	963	606	35	5,5 %	3,5 %
3 m	859	502	139	21,7 %	14,0 %

Tabelle 10: Vollständigkeit und Korrektheit der Objektklasse „große Bäume“

Größe Bäume (Kronenfläche $\geq 2 \text{ m}^2$)					
Fehler-radius	FP	FN	TP	Vollständigkeit (Producer's Accuracy)	Korrektheit (User's Accuracy)
1 m	1477	1788	413	18,8 %	21,9 %
2 m	904	1215	986	44,8 %	52,2 %
3 m	685	996	1205	54,8 %	63,8 %

6 Ergebnis des Massenprozessierungssystems

6.1 Aufbau des Massenprozessierungssystems

Das Massenprozessierungssystem besteht aus den beiden Subsystemen „Katalogisierung“ und „Verteiltes Rechnen“ (siehe Abbildung 40). Das System „Katalogisierung“ übernimmt die Reorganisation der Eingangsdaten. Ergebnis der Reorganisation sind die speziell an die Erfordernisse des Algorithmus angepassten Prozesseingangsdaten. Im System „Verteiltes Rechnen“ erfolgt die eigentliche Prozessierung sowie die Generierung der kundenspezifischen Produkte unter Einbeziehung der Informationen der Eingangsdatenstruktur, die in der Katalog-Datenbank abgelegt sind. Auf diese Weise kann das Prozessierungssystem weitestgehend von der kundenspezifischen Datenstruktur entkoppelt werden.

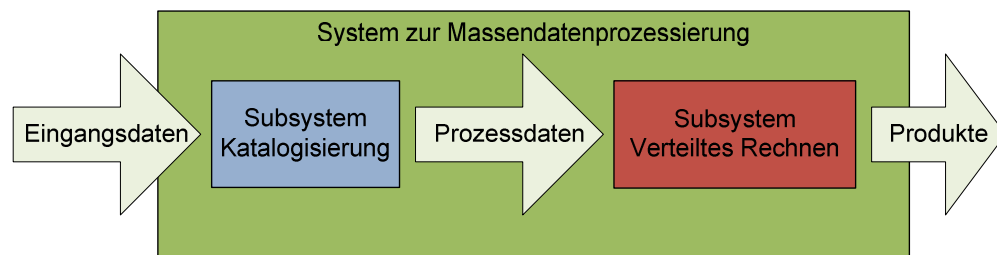


Abbildung 40: Prinzipieller Aufbau des Massenprozessierungssystems

6.2 Das Subsystem „Katalogisierung“

6.2.1 Struktur der Eingangsdaten

Die vom Projektpartner gelieferten Eingangsdaten werden in Prozesseingangsdaten, kurz Prozessdaten überführt. Im konkreten Fall wurden die LIDAR-Daten und Maskenpolygon-Shapes in Form von Kacheln geliefert. Eine Kachel deckt ein Gebiet von 2 km x 2 km ab und besteht aus vier ASCII-Dateien, die die LIDAR-Daten beinhalten sowie einem Polygon-Shape mit den Masken des Kontextmodells (siehe Kapitel 3.3). Die Kacheln sind vom Projektpartner nach dessen Notation geordnet und mit einer eindeutigen Nummer, der Kachelnummer, gekennzeichnet. Diese Kachelnummer ist jedoch nicht für eine Indizierung aus Sicht der Reorganisation (Merge der Überlappungsbereiche aus Nachbarkacheln) geeignet.

6.2.2 Optimierung von Überlappungsbereich und Prozesskachelgröße

Optimierung von Überlappungsbereich

Die Größe der Prozesskachel hat unmittelbaren Einfluss auf die Performance der Verarbeitung (vergleiche Kapitel 5.2.1). Andererseits müssen die Prozessdaten derart aufbereitet werden, dass zu extrahierende Objekte aufgrund einer Zerschneidung an den Kachelgrenzen nicht mehr eindeutig einer Prozesskachel zugeordnet werden können. Abbildung 41 verdeutlicht das Problem der Objektzerschneidung an der Kachelgrenze (Kachelgröße im Bild nicht maßstäblich). Wird die Prozesskachel um einen Überlappungsbereich ausgedehnt, so werden die Objekte nicht mehr zerschnitten. Die Größe des Überlappungsbereichs ist dabei von der maximalen Objektgröße abhängig.

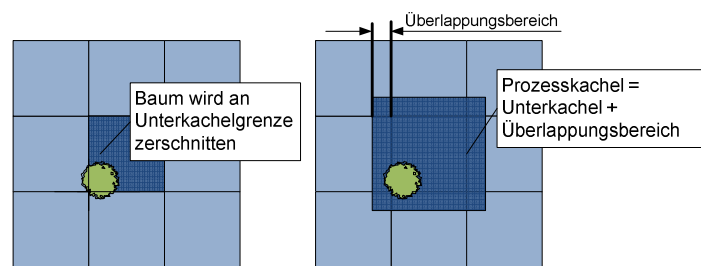


Abbildung 41: Überlappungsbereich löst das Problem der Objektzerschneidung

Im Folgenden wird zunächst die optimale Größe des Überlappungsbereichs anhand einer Voruntersuchung bestimmt. Dazu wurde mittels des implementierten Algorithmus eine vollständige Eingangskachel verarbeitet. Aus den extrahierten Einzelbaum-Objekten (~30.000 Stück) wurde in der Verteilung des Durchmessers der konvexen Hülle das 95%-Quantil bestimmt. Das Ergebnis lieferte einen Wert von $Q_{95} = 17,3$ m. Durch Aufrundung auf einen Radius von 10 m wurde ein sicheres Maß für den Überlappungsbereich definiert.

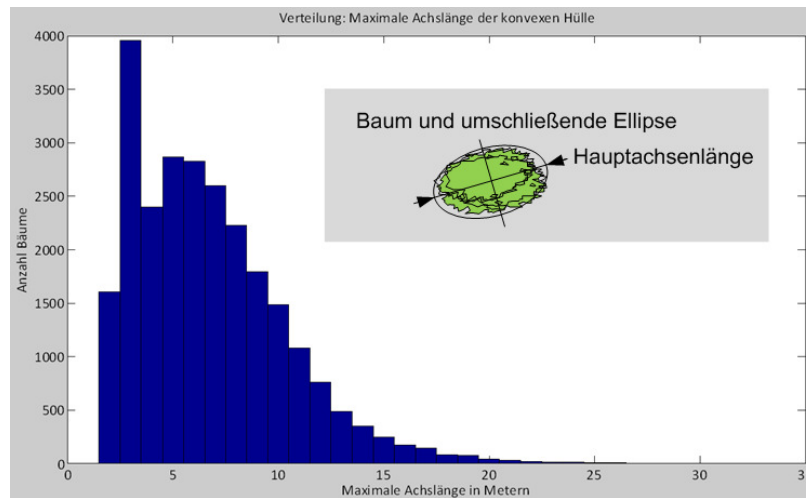


Abbildung 42: Voruntersuchung: Häufigkeitsverteilung des Durchmessers der konvexen Hülle

Bestimmung der optimalen Prozesskachelgröße

Die optimale Prozesskachelgröße hängt von zwei Faktoren ab, der Größe des Überlappungsbereichs und der Zeitkomplexität bzw. dem Zweitaufwand des Algorithmus.

Der Überlappungsbereich der Prozesskachel muss als nutzlose Fläche betrachtet werden, die jedoch maßgeblich zum Rechenaufwand beiträgt. Daher kann das Verhältnis zwischen effektiver Prozesskachelgröße (ohne Überlappungsbereich) und der Größe der Prozesskachel einschließlich des Überlappungsbereichs als Maß für die Effizienz des Algorithmus betrachtet werden.

$$c = \frac{a^2}{(a + 2b)^2}$$

c = Verhältnis effektive Prozesskachelgröße/ Gesamtkachelgröße

a = Kantenlänge der quadratischen Prozesskachel in Pixel

b = Breite des Überlappungsbereichs in Pixel

Formel 10: Maß für die Effizienz des Algorithmus abhängig von der Größe des Überlappungsbereichs

Das Maß für die Zeitkomplexität des Algorithmus ist proportional zur Zeitkomplexität des aufwendigsten Teilschritts. Dies ist im vorliegenden Fall die Berechnung der Delaunay-Triangulation mit einer Komplexität „ $O(n \times \ln(n))$ “, wobei „ n “ der Zahl der Knoten entspricht, die in die Berechnung der Delaunay-Triangulation einfließen. Diese Zahl (Anzahl der ALS-Datenpunkte) ist im vorliegenden Fall proportional zur Fläche der Prozesskachel in Pixel.

Überträgt man das Maß der Zeitkomplexität in die obere Gleichung, so erhält man das Verhältnis der Zeitkomplexität von effektiver Prozesskachelfläche zu Gesamtfläche.

Ersetzt man anschließend die Breite des Überlappungsbereichs mit dem zuvor definierten Wert „20“, so ergibt dies die in Abbildung 43 dargestellte Kennlinie, die das Maß der effektiven Zeitkomplexität über die Kantenlänge der Prozesskachel in Pixel aufträgt. Aus der Kennlinie kann interpretiert werden, dass bei geringer Kantenlänge ein großer Teil des Zeitaufwandes in die Verarbeitung des redundanten Überlappungsbereichs fließt. Erst ab einer Kantenlänge von mehr als 180 Pixel beträgt der Anteil des Zeitaufwands für die effektiven Prozesskachelfläche mehr als 80 %. Darüber hinaus ändert sich das Verhältnis bei steigender Kantenlänge nur noch unwesentlich. Um gleichzeitig im Rahmen der Entwicklung mit handhabbaren Testdaten zu arbeiten, wurde anhand dieses Ergebnisses eine effektive Prozesskachelgröße von 200 Pixel x 200 Pixel definiert.

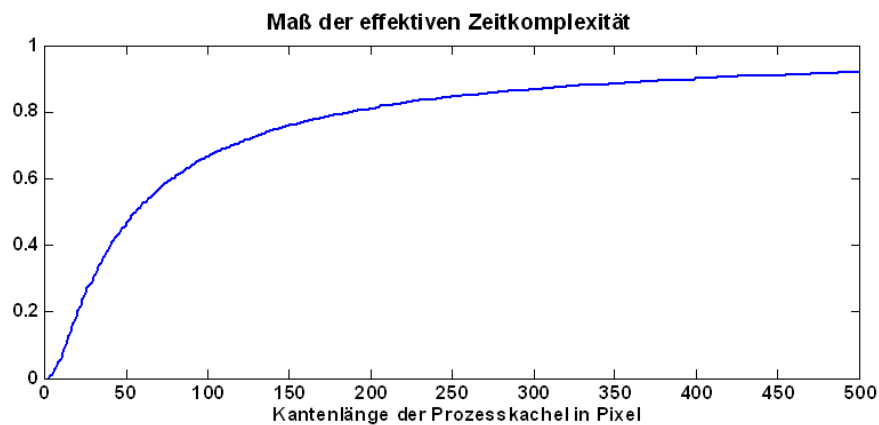


Abbildung 43: Maß der effektiven Zeitkomplexität

6.2.3 Von den Eingangsdaten zu den Prozessdaten

Dem Untersuchungsgebiet wurde ein reguläres, virtuelles Gitter von 1300 x 1300 Zellen mit einer Kantenlänge von 200 m überlagert (Kantenlänge der Prozesskachel = 200 Pixel; Rasterweite der Oberflächengenerierung = 1 m). Dieses Gitter war Grundlage der Zerlegung der ALS-Daten und Maskenpolygone in Prozessdaten. Abbildung 44 veranschaulicht dies durch einen Zoom vom Untersuchungsgebiet (der Rahmen des regulären Gitters ist als Prozesskachelrahmen dargestellt) in einen Ausschnitt der Eingangsdaten (grün) bis auf die Ebene der Prozessdaten (blau).

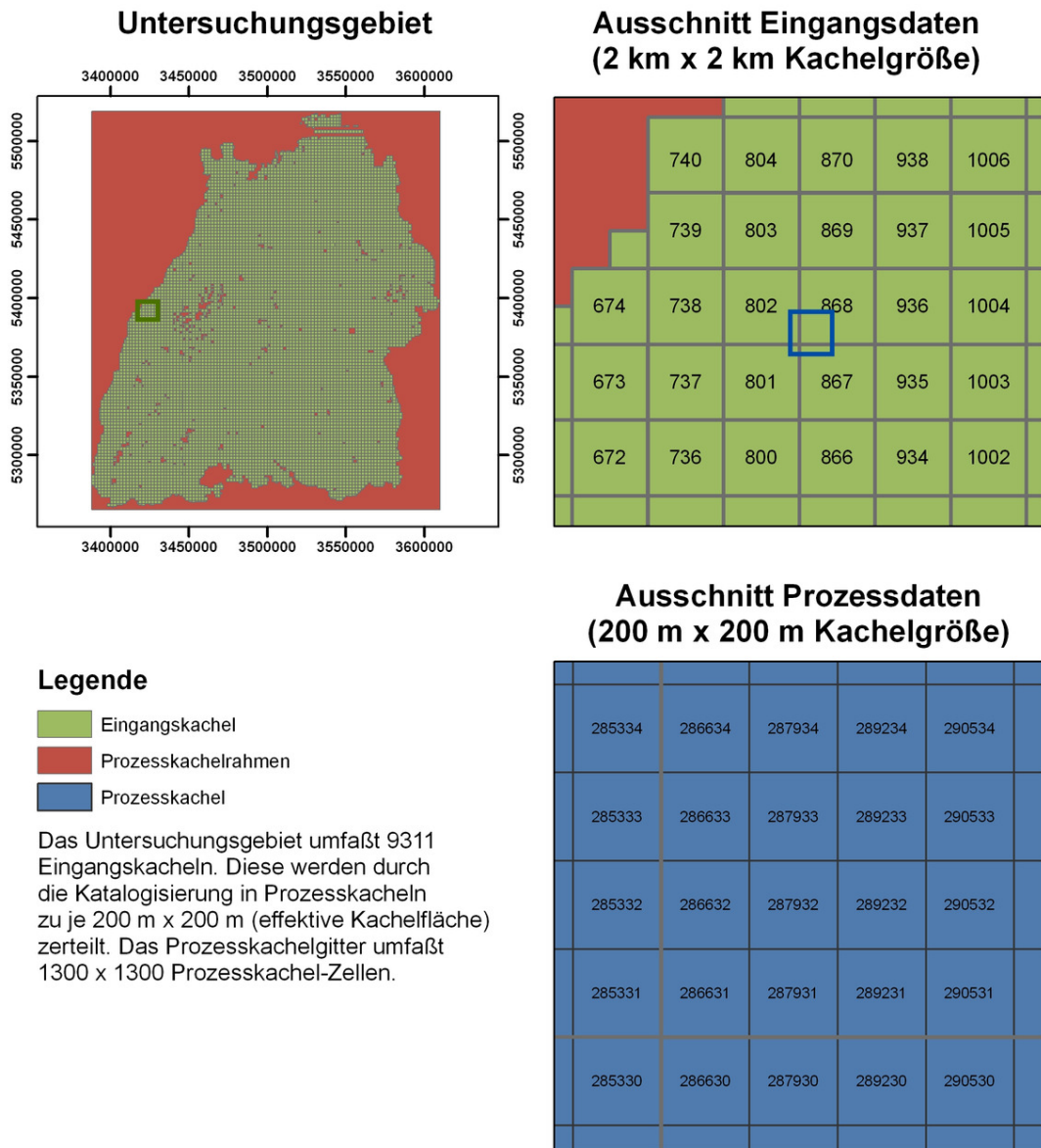


Abbildung 44: Zoom vom Untersuchungsgebiet in die Eingangsdaten und von dort in die Prozessdatenebene

Der Ursprung des regulären Gitters wurde in den Ursprung des Untersuchungsgebiets gelegt. Die Indizierung der Gitterzellen erfolgte derart, dass zu jeder Zelle bzw. Prozesskachel alle Nachbarzellen arithmetisch bestimmt werden können (siehe Abbildung 46).

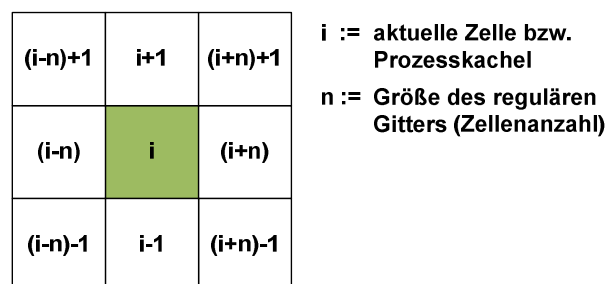


Abbildung 45: Adressierung der Nachbarzellen im regulären Gitter

Bei der Ablage der reorganisierten Prozesskacheln wird ein pragmatischer Weg gewählt. Die Daten jeder Prozesskachel werden in je einem eigenen Unterverzeichnis gespeichert. Da jedoch die Ablage einer derart großen Menge Unterverzeichnisse in einem einzigen Projektverzeichnis erhebliche Performanceeinbußen aufgrund der Zugriffszeiten über das Filesystem bedeuten würde, wird eine hierarchische Verzeichnisstruktur gewählt. Dabei dient die Prozesskachelkennung als Zugriffsschlüssel. Die sechsstellige numerische Prozesskachelkennung wurde in Zweierblöcke zu je zwei Ziffern zerlegt. Jeder Zweierblock bildet ein Unterverzeichnis in der Verzeichnis-Hierarchie. So werden beispielsweise die Daten der Prozesskachel „287931“ im Unterverzeichnis „\<Prozessdaten>\28\79\31“ gespeichert. Die Daten der benachbarten Prozesskachel „287932“ werden im Unterverzeichnis „\<Prozessdaten>\28\79\32“ gespeichert. Abbildung 46 veranschaulicht die Zuordnung zwischen Eingangsdaten- und Prozessdatenablage. Zentrales Element der Darstellung ist die Katalog-Datenbank, in der die Zuordnung zwischen Eingangs- und Prozessdaten gespeichert wird. Der Aufbau Katalog-Datenbank ist Inhalt des folgenden Kapitels.

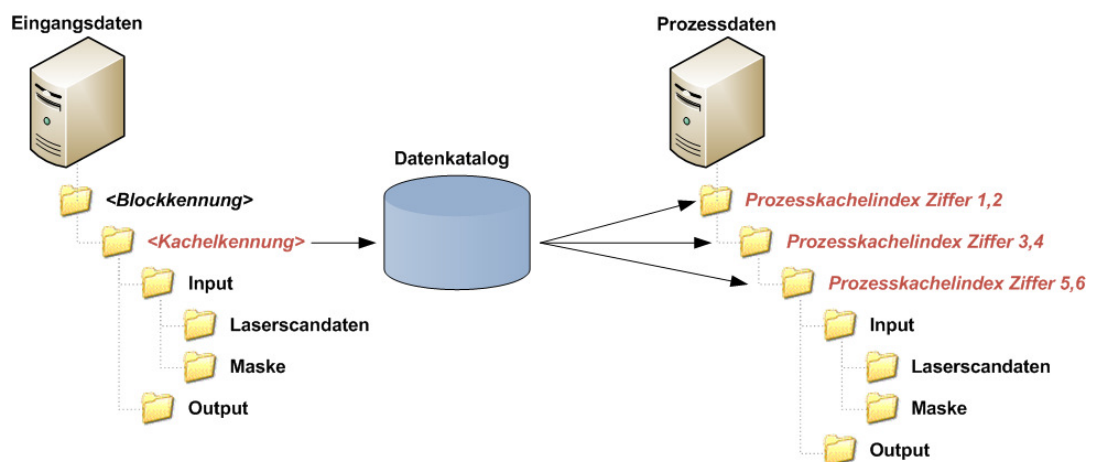


Abbildung 46: Datenablage von Eingangs- und Prozessdaten

6.2.4 Datenbankschema des Katalogs

In der Katalogdatenbank werden die Eingangsdaten mit den Prozessdaten verknüpft. Das Entity-Relationship-Diagramm (ER-Diagramm) in Abbildung 47 zeigt, dass die Verknüpfung über Fremdschlüssel-Beziehungen (Foreign Keys, FK) realisiert ist. Die Tabelle „*ProjectSettings*“ ergänzt das Schema um eine Tabelle, in der mittels Schlüssel-Werte-Paaren die Konfigurationsparameter (siehe 5.1.2) des Projekts hinterlegt sind.

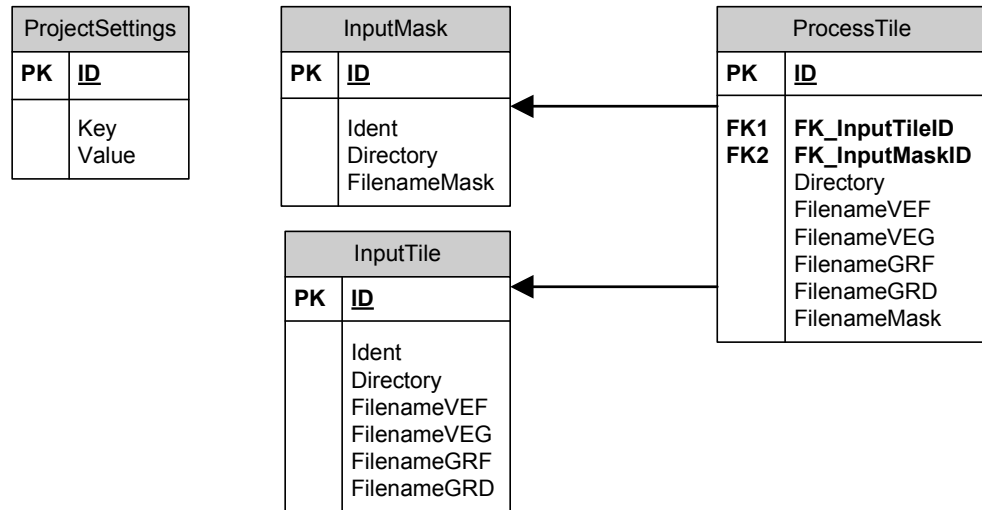


Abbildung 47: ER-Diagramm des Katalogdatenbankschemas

6.2.5 Ablauf der Reorganisation

Die Reorganisation der Eingangsdaten wurde als Python-Skript unter Verwendung der Bibliotheken GDAL und OGR realisiert. In einer ersten Schleife über alle Eingangsdaten werden diese zunächst mittels Clipping in Prozessdaten zerlegt (ALS-Daten und Masken-Shapefiles). Die Informationen der Eingangs- und Prozessdaten werden in die Katalogdatenbank geschrieben. In einer zweiten Schleife über alle generierten Prozessdaten werden je Prozesskachel die Überlappungsbereiche der Nachbarkacheln bestimmt und zur jeweiligen Prozesskachel hinzugefügt. Abbildung 48 veranschaulicht diesen Ablauf nochmals in Form eines vereinfachten Aktivitätsdiagramms. Dabei wird auf die detaillierte Beschreibung aller Einzelschritte verzichtet.

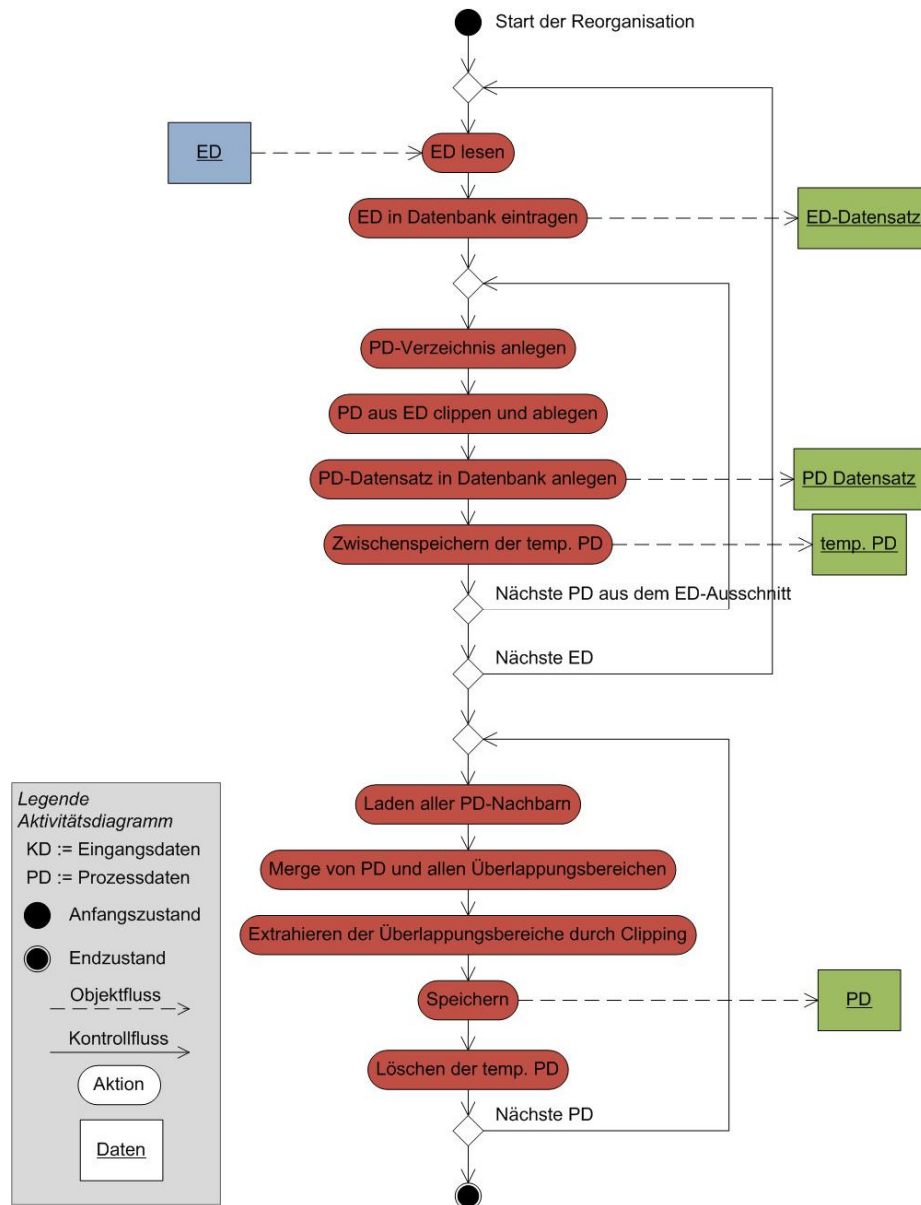


Abbildung 48: Vereinfachter Ablauf der Reorganisation

6.3 Das Subsystem „Verteiltes Rechnen“

6.3.1 Systemaufbau

Das für die Untersuchung aufgebaute System besteht aus den folgenden Komponenten (vergleiche Abbildung 49):

- Einem BOINC-Server mit allen dazu gehörigen Komponenten (siehe Kapitel 2.6) in Form einer Virtual Machine, die auf einem Virtualisierungssystem (VMware) eingerichtet wurde (Intel Dual Core 2.4 GHz, Scientific Linux 5.2).
- Einem Network Attached Storage-Server (NAS), auf dem die Eingangs- und Prozessdaten abgelegt sind. Die Ergebnisdaten des Algorithmus werden ebenfalls auf dem NAS abgelegt.
- Vier Desktop-Rechnern (Intel Dual Core 2.4 GHz, Windows XP Professional)

Diese Testumgebung mit vier Desktop-Rechnern wird so gewählt, dass die Performance des Systems derjenigen des Vergleichssystems (siehe Kapitel 6.4.1) gegenüber gestellt werden konnte.

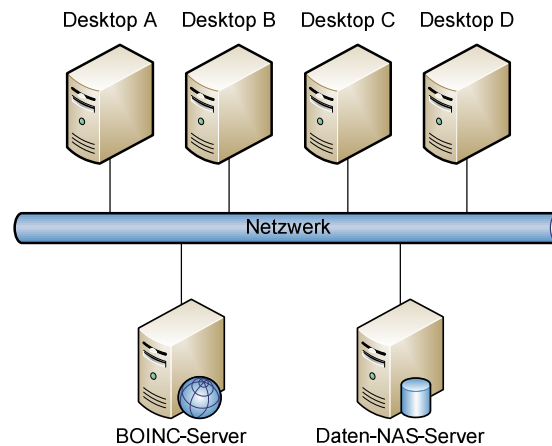


Abbildung 49: Systemaufbau

6.3.2 Systemkonfiguration

Generierung der Jobs

In jedem Job wird eine gesamte Eingangskachel, d.h. 100 Prozesskacheln, verarbeitet. Die Jobgenerierung wurde als Python-Skript implementiert. In einer Schleife über alle Eingangskachel-Einträge in der Katalogdatenbank wurde je Eingangskachel mit Hilfe des Kommandos „*create_work*“ aus der Werkzeugsammlung des BOINC-Systems ein Eintrag in der Jobqueue des BOINC-Servers generiert. Die Jobqueue ist die Schnittstelle zwischen der projektspezifischen Jobgenerierung und dem Scheduler des BOINC-Systems (vergleiche Kapitel 2.6).

Zentrale Ablage der Projektanwendung

Die Projektanwendung besteht aus einer zentral abgelegten Batchdatei. Diese Batchdatei enthält lediglich den Programmaufruf des Algorithmus „*orchard.exe*“ wobei die Übergabeparameter des BOINC-Jobs (Prozesskachelkennung) an den Algorithmus weitergegeben werden. Weiterhin wird die Wrapperanwendung sowie die Konfiguration der Wrapperanwendung an zentraler Stelle im Netzwerk (NAS-Server) abgelegt.

6.4 Bewertung des Massenprozessierungssystems

6.4.1 Vergleichssystem

Umsetzung des Vergleichssystem

Das Vergleichssystem basiert auf einem Job, der in Python implementiert ist. Dieser Job startet die Verarbeitung einer Eingangskachel, indem er die Anwendung (den Algorithmus) startet und eine Eingangskachelkennung übergibt. Zur vollen Auslastung des Vergleichssystems werden acht Jobs gleichzeitig gestartet. Damit zwei Jobs nicht dieselbe Kachel bearbeiten, wird ein rudimentärer Verriegelungsmechanismus implementiert. Dieser Verriegelungsmechanismus basiert auf dem Prinzip einer zugriffsgesteuerten Queue. Im vorliegenden Fall entspricht die Queue einer ASCII-Datei, in der alle zu verarbeitenden Kachelkennungen aufgelistet sind. Die Transaktionssteuerung, die Teil des Jobskripts ist, stellt wie folgt sicher, dass immer nur ein Job gleichzeitig auf die Queue zugreifen kann. Bevor ein Job auf die Queue zugreifen kann, legt er eine temporäre Datei (Lockdatei) an. Sobald diese Lockdatei existiert, kann der Job auf die Queue zugreifen, die nächste Kachelkennung auslesen und aus der Queue löschen. Anschließend löscht der Job die temporäre Lockdatei. Möchte gleichzeitig ein anderer Job auf die Queue zugreifen, wird dieser blockiert, solange die temporäre Lockdatei existiert. Dieser Ablauf ist nochmals in Abbildung 50 anhand eines Aktivitätsdiagramms veranschaulicht.

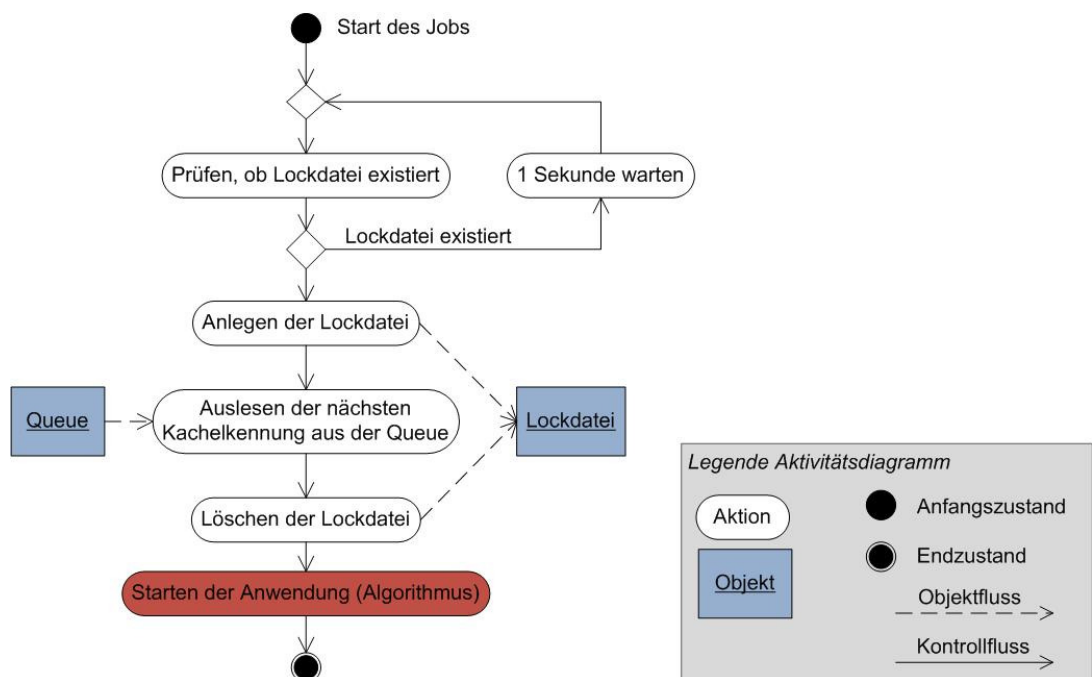


Abbildung 50: Verriegelungsmechanismus des Vergleichssystems

Die Anwendung bearbeitet sequentiell alle Prozesskacheln der übergebenen Eingangskachel, indem jeweils alle kachelspezifischen Eingangsdaten (ALS-Daten und Maskenshape, Ablageverzeichnis der Ergebnisdateien) ermittelt werden. Je Prozesskachel führt die Vergleichsanwendung den Einzelbaumextraktionsalgorithmus „orchard.exe“ aus. Der Algorithmus speichert die Ergebnisdateien im übergebenen Ablageverzeichnis (Baumposition und Baumkontur als Shapefile).

6.4.2 Ergebnis des Vergleichs

Der Vergleich stellt die Laufzeit der Bearbeitung von je 500 Eingangskacheln im BOINC-System und im Vergleichssystem gegenüber. Bei beiden Systemen wurde jeweils die Laufzeit vom Start des Jobs (vor dem Laden der Eingangsdaten) bis zu dessen Ende (nach dem Speichern der Ausgangsdaten) gemessen. Aus der Häufigkeitsverteilung der Laufzeiten (siehe Abbildung 51) ergibt sich für die Job-Laufzeit auf dem BOINC-System ein Mittelwert von **28,4 Minuten** und für die Job-Laufzeit auf dem Vergleichssystem ein Mittelwert von **26,0 Minuten**.

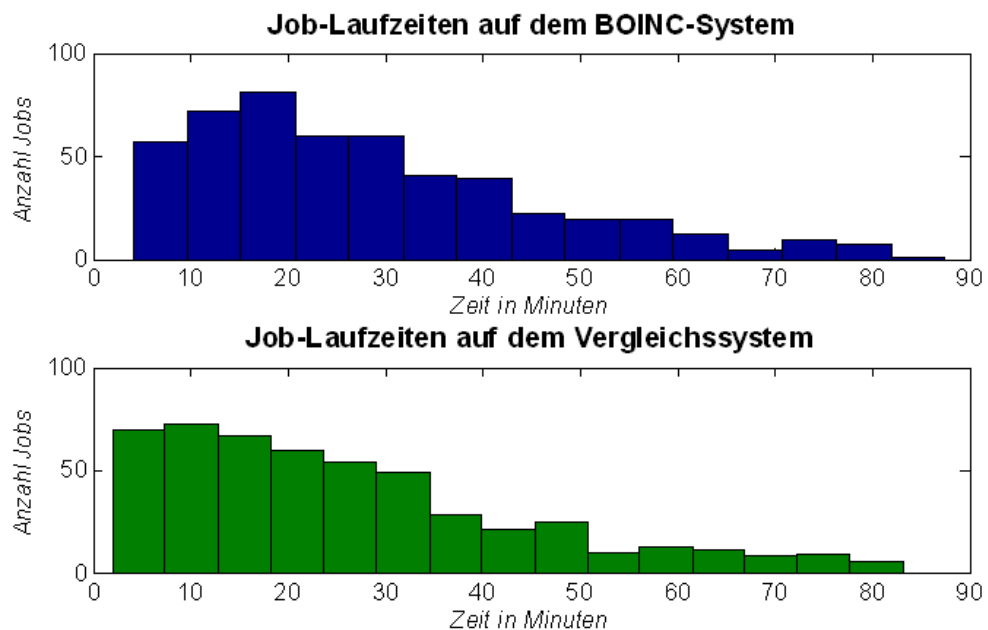


Abbildung 51: Häufigkeitsverteilung der Laufzeiten im BOINC-System und im Vergleichssystem

7 Diskussion und Ausblick

7.1 Diskussion

Die Untersuchung der beiden dieser Arbeit zugrunde liegenden Fragestellungen hat zu folgenden Ergebnissen geführt. Es konnte gezeigt werden, dass es auf der Basis der verfügbaren ALS-Daten und eines Kontextmodells prinzipiell möglich ist, Aussagen über den Streuobstbaumbestand in Baden-Württemberg zu treffen. Desweiteren hat die Untersuchung des Problems der Massendatenverarbeitung zu dem Ergebnis geführt, dass die Verwendung eines Desktop Grid Computing System für die Verarbeitung großer Mengen von Fernerkundungsdaten nur bedingt geeignet ist.

Datengrundlage

Die verwendeten ALS-Daten stammen aus einer mehrjährigen Befliegungskampagne zur Erstellung eines neuen DGM des Landes Baden-Württemberg. Im Kontext dieser Anwendung kann von hochaufgelösten Daten gesprochen werden. Wie jedoch auch Blaschke (Blaschke, Burnett & Pekkarinen 2004) verdeutlicht hat, besteht ein enger Zusammenhang zwischen der Größe der zu extrahierenden Objekte und der erforderlichen Auflösung der Bilddaten (vergleiche Kapitel 2.2.2). Auf die Extraktion von Einzelbäumen aus den verwendete ALS-Daten bezogen bedeutet dies, dass die Auflösung der ALS-Daten in einem Bereich liegt, in dem es schwierig ist, die geforderten Objekte zuverlässig voneinander abzugrenzen und zu extrahieren. Besonders erschwert wird diese Problematik durch die Tatsache, dass die Befliegung der ALS-Daten in den vegetationsarmen Wintermonaten stattgefunden hat. Dennoch konnte auf Grundlage der Erkenntnis aus früheren Arbeiten zum Thema Einzelbaumextraktion aus ALS-Daten ein speziell auf die Daten abgestimmter Algorithmus entwickelt werden, der eine Abschätzung des aktuellen Streuobstbaumbestandes in Baden-Württemberg ermöglicht. Die Verfügbarkeit weiterer Informationen zu den ALS-Daten, wie die Reflexionsintensität, wäre bei der Untersuchung in jedem Fall hilfreich gewesen. Diese Daten standen jedoch nicht zur Verfügung.

Methode

Ausgehend von der Entwicklung eines Objektmodells wurde ein, speziell auf die vorliegenden ALS-Daten abgestimmter Algorithmus entwickelt.

Der Ansatz, einer speziell auf die Daten abgestimmten Entwicklung, steht dabei im Gegensatz zu den Überlegungen von Bechtel (Bechtel 2007), der die Realisierung eines generalisierten Expertensystems propagiert. Ein solches Expertensystem kann auf hochaufgelöste Daten angewendet werden, bei unzureichender Datengrundlage wie im vorliegenden Fall, wird es jedoch weiterhin notwendig sein, speziell angepasste Verfahren zu entwickeln.

Algorithmus im Detail

Die verwendete Methode der Rasterung von vektoriellen ALS-Punktwolken zur Generierung eines digitalen Gelände- und Oberflächenmodells basiert auf der Nearest-Neighbor-Interpolation. Dieses Interpolationsverfahren hat den Vorteil, dass es sehr performant ist. Andererseits birgt es den Nachteil, dass Fehler in den Ausgangsdaten ungefiltert in die Oberflächengenerierung einfließen. Fehler werden also nicht geglättet, wie dies bei approximativen Interpolationsverfahren der Fall ist.

Bei der Objektextraktion von Einzelbäumen aus ALS-Daten hat die Watershed-Transformation (siehe Kapitel 2.4.2), zu den besten Ergebnissen geführt. Bei dem im Vorfeld der Arbeit erprobten Region-Growing Verfahren konnten trotz der Untersuchung verschiedener Methoden der Saatpunkte-Generierung sowie unterschiedlicher Abbruchkriterien, keine befriedigenden Ergebnisse erzielt werden.

Validierung

Da bis zum Abschluss der Arbeit keine verwertbaren Ground Truth Daten vorlagen, musste bei der Validierung auf verfügbare Orthophotos zurückgegriffen werden. Da die Befliegungen der ALS-Daten und der Orthophotos jedoch bis zu 5 Jahre auseinanderlagen, bestehen teilweise erhebliche Unterschiede in den Daten. Wie in Kapitel 3.1.4 gezeigt wird, fehlen auf den Orthophotos ganze Flächen an Streuobstbeständen, die in den 5 Jahre älteren ALS-Daten noch vorhanden sind. Andererseits sind auf den Orthophotos Neubestände (Niedrigstämme) zu erkennen, die 5 Jahre zuvor entweder noch nicht existierten oder zwischenzeitlich frühere Altbestände ersetzt haben. Eine Unterscheidung der Bestände im Rahmen der Digitalisierung der Referenzdaten war aufgrund fehlender Informationen nicht möglich. Diese Tatsache erschwert die Aussagekraft der Validierung. Konzentriert man die visuelle Auswertung jedoch auf markante Einzelbäume, so kann festgehalten werden, dass der Algorithmus grundsätzlich gute Ergebnisse liefert.

Aufbau eines Massenprozessierungssystems

Im Fokus der Untersuchung des Problems der Massendatenverarbeitung in der Fernerkundung stand nicht die Konzeption und der Aufbau eines Hochleistungsrechners aus dem Bereich des Cluster Computings, sondern die Konzeption eines Systems, das die bestehende Hardwareressourcen und die bestehende Netzwerkinfrastruktur eines KMUs nutzen kann, um rechenintensive Bildanalyseaufgaben zu bearbeiten.

Das verwendete System BOINC zeichnet sich nicht nur durch seine flexible Anpassung an anwendungsspezifische Erfordernisse aus, sondern erfüllt auch die in Zusammenarbeit mit dem Projektpartner erstellten Anforderungen und Rahmenbedingungen im Vergleich zu Alternativsystemen am besten. Nachteilig ist die, auf Grund der gewählten Systemarchitektur (siehe Kapitel 6.3.1), zeitintensive Verteilung der Daten auf den Clientrechnern. Das aufgebaute System wurde speziell für die Bearbeitung sporadischer Projektaufträge konzipiert, in denen jeweils große Datenmengen mittels projektspezifischer Workflows zu verarbeiten sind. Daher wurde auf die Einrichtung eines eigenen Hochgeschwindigkeitsrechner (Cluster) verzichtet.

Reorganisation und Katalogisierung

Die Entkopplung der verteilten Verarbeitung von den Eingangsdaten durch den Aufbau eines Subsystems zur Reorganisation und Katalogisierung hat sich als sehr hilfreich erwiesen. Die Reorganisation der Daten konnte dazu genutzt werden, das Problem der Zerschneidung von Objekten an Kachelgrenzen dadurch zu lösen, dass zu den Daten jeder Kachel die Daten eines Überlappungsbereichs zu den Nachbarkacheln hinzugefügt wurden.

7.2 Ausblick

Der entwickelte Algorithmus zur Einzelbaumextraktion wurde speziell auf die vorliegenden ALS-Daten abgestimmt und kann nur durch projektspezifische Anpassungen auf andere Datenbestände übertragen werden. Durch eine Erweiterung der Merkmalsextraktion um Texturparameter, wie beispielsweise Rauheit, sowie eine Klassifikation mittels Clusteranalyse oder Fuzzy-Klassifikator könnte das Objektmodell verfeinert werden. Es könnte zudem untersucht werden, welche Merkmale dazu beitragen, Klassen wie Sträucher, Niedrig- und Hochstämme oder Hecken voneinander zu trennen.

Segmentierungsverfahren

Das verwendete Segmentierungsverfahrens mittels Watershed-Transformation hat sich nicht nur in der vorliegenden Arbeit, sondern bereits in früheren Arbeiten zur Einzelbaumextraktion bewährt (siehe Kapitel 2.4.2). Dennoch wäre eine Untersuchung alternativer Segmentierungsverfahren, wie beispielsweise des Hill-Climbing-Algorithmus (vergleiche Heurich et al. 2004), der morphologische Top-Hat-Transformation (vergleiche Andersen, Reutebuch & Schreuder 2001), des Felzenszwalb-Huttenlocher-Algorithmus, des Active Shape Models (eine Erweiterung der aktiven Konturen) oder einer Kantendetektion auf der Basis von Multiagenten-Algorithmien sehr interessant.

Kombination von Laser Scan- und mit Spektralen Daten

Wie in Kapitel 2.4.2 aufgezeigt, zeichnet sich eine methodische Entwicklung ab, bei der ALS-Daten mit spektralen Daten (multispektral, hyperspektral) kombiniert werden. So hat Diedershagen (Diedershagen O. 2003) beispielsweise in seiner Arbeit ALS-Daten mit Farbfotos und Kontextinformationen aus einem forstwirtschaftlichen Geoinformationssystem kombiniert. Eine solche Kombination von ALS-Daten und zusätzlichen spektralen Informationen könnte auch im vorliegenden Fall die Informationsdichte erhöhen und somit zur Verbesserung der Segmentierungsergebnisse und Merkmalsextraktion beitragen.

Grid- und Clustertechnologien

Im Bereich der Grid-Technologien, insbesondere im Umfeld von E-Science, sind derzeit verschiedenen Trends zu beobachten. Eine Untersuchung der verschiedenen Entwicklungen und deren Anwendbarkeit auf Fragestellungen der Fernerkundung und Geoinformatik wären, aufgrund immer umfangreicherer Datengrundlagen und immer komplexer werdender Analyse- und Auswertungsverfahren, von großem Interesse. Als Beispiel sei hier das Projekt GDI-Grid angeführt (Neuroth, Kerzel & Gentzsch 2007).

Andererseits schreitet auch die Entwicklung der parallelen Programmierung, insbesondere der GPU-Programmierung (Graphics Processing Unit) voran. Auch hier wäre eine Untersuchung hinsichtlich der Anwendbarkeit auf fernerkundliche Bildanalyseverfahren interessanter Gegenstand einer Untersuchung.

8 Literatur und Quellen

8.1 Literaturverzeichnis

AdV 2003, *ATKIS-Objektkatalog. Teilkatalog ATKIS-Basis-DLM*.

Albertz, Jörg 2007, *Einführung in die Fernerkundung. Grundlagen der Interpretation von Luft- und Satellitenbildern*, Wiss. Buchges., Darmstadt. Online verfügbar unter: <http://www.gbv.de/dms/hebis-darmstadt/toc/185831990.pdf> [20 Oktober 2009].

Andersen, Hans-Erik, Reutebuch, Stephen E. & Schreuder, Gerard F. 2001, 'Automated Individual Tree Measurement through Morphological Analysis of a LIDAR-based Canopy Surface Model' in *Proceedings of the first International Precision Forestry Cooperative Symposium*, pp. 11–22.

Anderson, David P., Cobb, Jeff, Korpela, Eric, Lebofsky, Matt & Werthimer, Dan 2002, 'SETI@home: an experiment in public-resource computing', *Commun. ACM*, vol. 45, no. 11, pp. 56-61.

Baatz, M. & Schäpe, A. 2000, 'Multiresolution segmentation - an optimization approach for high quality multi-scale image segmentation'. *Angewandte Geographische Informationsverarbeitung XII. Beiträge zum AGIT-Symposium Salzburg 2000*, eds Strobl & Blaschke, Wichmann, Heidelberg.

Barber, C. Bradford, Dobkin, David P. & Huhdanpaa, Hannu 1996, 'The quickhull algorithm for convex hulls', *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469-483.

Bauke, H & Mertens, S 2006, *Cluster Computing. Praktische Einführung in das Hochleistungsrechnen auf Linux-Clustern*. Online verfügbar unter: <http://www.zentralblatt-math.org/zmath/en/search/?an=1095.68540> [20 Oktober 2009].

Baumgartner, Albert et al 1999, 'Automatic Road Extraction Based on Multi-Scale, Grouping, and Context'.

Bechtel, B 2007, *Objektextraktion von Bäumen aus Luftbildern. Vergleich und Steuerung von Segmentierungsverfahren zur Vorbereitung eines Expertensystems*, Institut für Geographie, Universität Hamburg. Online verfügbar unter: http://www.uni-hamburg.de/geographie/personal/Mitarbeiter/bechtel/index_/da_bechtel_final_screen.pdf [18 August 2009].

Bengel, G, Baun, C, Kunze, M & Stucky, K 2008, *Masterkurs Parallele und Verteilte Systeme. Grundlagen und Programmierung von Multicoreprozessoren, Multiprozessoren, Cluster und Grid*. Online verfügbar unter: <http://dx.doi.org/10.1007/978-3-8348-9516-5> [20 Oktober 2009].

Blaschke, T 2000, *Objektextraktion und regelbasierte Klassifikation von Fernerkundungsdaten: Neue Möglichkeiten für GIS-Anwender und Planer. Möglichkeiten für GIS-Anwender und Planer*. CORP 2000.

Blaschke, Thomas, Burnett, Charles & Pekkarinen, Anssi 2004, 'Image Segmentation Methods for Object-based Analysis and Classification' in *Remote Sensing Image Analysis: Including The Spatial Domain*, Kluwer Acad. Publishers, pp. 211–236.

Bonn, M 2008, *JoSchKa. Jobverteilung in heterogenen und unzuverlässigen Umgebungen*. Univ., Diss--Karlsruhe, 2008. Online verfügbar unter: <http://digbib.ubka.uni-karlsruhe.de/volltexte/1000009291> [20 Oktober 2009].

Brandtberger, T. & Walter, F. 1998, 'Automated delineation of individual tree crown in high spatial resolution aerial images by multiple-scale analysis' in *Machine Vision and Applications*, pp. 64–73.

- Bubak, Marian & Albada, Geert Dick van 2004, *Computational science. : ICCS 2004, 4th international conference, Kraków, Poland, June 6 - 9, 2004; proceedings*, Springer, Berlin. Online verfügbar unter: <http://www.zentralblatt-math.org/zmath/en/search/?an=1051.68007> [20 Oktober 2009].
- Bundesamt für Kartographie und Geodäsie 2008, *Digitales Basis-Landschaftsmodell. Basis-DLM*, Frankfurt am Main. Online verfügbar unter: www.geodatenzentrum.de [03 September 2009].
- Bunse, Christian & Knethen, Antje von 2008, *Vorgehensmodelle kompakt*, Spektrum Akad. Verl., Heidelberg. Online verfügbar unter: http://deposit.d-nb.de/cgi-bin/dokserv?id=3038438&prov=M&dok_var=1&dok_ext=htm [20 Oktober 2009].
- Campbell, James B 2007, *Introduction to remote sensing*, Guilford Press, New York. Online verfügbar unter: <http://www.loc.gov/catdir/enhancements/fy0701/2006022914-b.html> [20 Oktober 2009].
- Chang, Anjin, Kim, Jung Ok, Ryu, Kiyun & Kim, Yong Il 2008, 'Comparison of methods to estimate individual tree attributes using color aerial photographs and LiDAR data', *WSEAS Trans. Sig. Proc.*, vol. 4, no. 1, pp. 21-27.
- Congalton, Russell G. & Green, Kass 1999, *Assessing the accuracy of remotely sensed data. Principles and practices*, Lewis Publishers, Boca Raton, FL. Online verfügbar unter: <http://www.envirobase.com/pdf/enb/L986/toc.pdf> / <http://www.ethbib.ethz.ch/readme/crc-text.html> [20 Oktober 2009].
- Diedershagen O., Koch B. Weinacker H. 2003, 'Automatic Estimation Of Forest Inventory Parameters Based On Lidar Multi Spectral And Fogis Data Proceedings Optical 3 D' in *Proceedings Optical 3-D Measurement Techniques VI*, pp. 259–269.
- Foody, Giles M & Atkinson, Peter M 2002, *Uncertainty in remote sensing and GIS*, Wiley, Chichester. Online verfügbar unter: <http://www.loc.gov/catdir/description/wiley036/2002191038.html> [20 Oktober 2009].
- Foster, Ian & Kesselman, Carl 1999, *The grid: blueprint for a new computing infrastructure*, Kaufmann, San Francisco, Calif. Online verfügbar unter: <http://www.loc.gov/catdir/description/els033/98030570.html> / <http://www.loc.gov/catdir/toc/els032/98030570.html> [20 Oktober 2009].
- Girstenbreu, Wolfgang 2006, *Streuobst in der Kulturlandschaft. Grenzüberschreitende Fachtagung des Instituts für Agrarökologie Ökologischen Landbau und Bodenschutz zusammen mit der Bayerischen Landesanstalt für Weinbau und Gartenbau am 11. - 12. Mai 2006 in Halsbach Waldbühne Landkreis Altötting; Tagungsband*, Bayerische Landesanstalt für Landwirtschaft, Freising.
- Gonzalez, Rafael C. & Woods, Richard E. 2006, *Digital Image Processing (3rd Edition)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Gougeon, F. A. 1995, 'A crown-following approach to the automatic delineation of individual tree crowns in high spatial resolution aerial images.' in *Canadian Journal of Remote Sensing*, pp. 274–284. Online verfügbar unter: http://bookstore.cfs.nrcan.gc.ca/detail_e.php?catalog=4583 [20 Oktober 2009].
- Hailer, Jörg 2005, *Erprobung von Laserscandaten zur Erfassung der forstlichen Erschließungseinrichtung am Beispiel des Distrikt I, Stadtwald Rottenburg*. Diplomarbeit, Rottenburg.
- Heurich, M., Kennel, E. & Kennel, G. 2007, 'Projekt Fernerkundung für Waldinventuren erfolgreich', *DerWald: Allgemeine Forst Zeitschrift für Wald und Umweltvorsorge*, no. 2, p. 70.
- Heurich, M., Persson, Ä., Holmgren, J. & Kennel, E. 2004, 'Detecting and Measuring Individual Trees with Laser-Scanning in Mixed Mountain Forest of Central Europe Using an Algorithm

- Developed for Swedish Boreal Forest Conditions' in *International Archives of Photogrammetry*, pp. 307–312.
- Hinz, Stefan 2004, *Automatische Extraktion urbaner Straßennetze aus Luftbildern*. Zugl.: München, Techn. Univ., Diss., 2003, Verl. der Bayerischen Akad. der Wiss. [u.a.], München.
- Hyypä, J. & Inkinen, M. 1999, 'Detecting and Estimating Attributes for Single Trees Using Laser Scanner' in *The Photogrammetric Journal of Finland*, pp. 27–42.
- Hyypä, J., Schardt, M., Haggrén, H., Koch, B. & Lohr, U. 1999, 'HIGH-SCAN. The First European-Wide Attempt to Derive Single-Tree Information from Laserscanner Data' in *The Photogrammetric Journal of Finland*, pp. 58–68.
- Jähne, Bernd 2005, *Digitale Bildverarbeitung. Mit ... 155 Übungsaufgaben und CD-ROM*, Springer, Berlin. Online verfügbar unter: <http://www.gbv.de/dms/hebis-darmstadt/toc/128384565.pdf> [20 Oktober 2009].
- Klauser, Berthold 2008, 'Das Digitale Geländemodell Baden-Württemberg. Landesvermessungsamt Baden-Württemberg' in *Forum Geoinformation Airborne Laserscanning*.
- Koch, B., Straub, C., Dees, M., Wang, Y. & Weinacker, H. 2009, 'Airborne laser data for stand delineation and information extraction', *Int. J. Remote Sens.*, vol. 30, no. 4, pp. 935-963.
- Kok, Roeland de 2001, *Objektorientierte Bildanalyse. Ein Lösungsansatz für den automatisierten Einsatz sehr hoch auflösender Satellitendaten für forstliche Fragestellungen*. München, Techn. Univ., Diss., 2001, [s.n.], S.I. Online verfügbar unter: <http://deposit.ddb.de/cgi-bin/dokserv?idn=964411032> [20 Oktober 2009].
- Landesvermessungsamt Baden-Württemberg 2009a, *ATKIS Basis-DLM*, Landesvermessungsamt Baden-Württemberg. Online verfügbar unter: http://www.lv-bw.de/lvshop2/ProduktInfo/geodaten/g_lvtla_atkis-dlm.htm [20 Oktober 2009].
- Landesvermessungsamt Baden-Württemberg 2009b, *Automatisierte Liegenschaftskarte ALK*, Landesvermessungsamt Baden-Württemberg. Online verfügbar unter: http://www.lv-bw.de/LVShop2/produktinfo/geodaten/g_lika_alb-alk.htm [20 Oktober 2009].
- Larsen, Morten & Rudemo, Mats 1998, 'Optimizing templates for finding trees in aerial photographs', *Pattern Recogn. Lett.*, vol. 19, no. 12, pp. 1153-1162.
- Lillesand, Thomas M., Kiefer, Ralph W. & Chipman, Jonathan W. 2008, *Remote sensing and image interpretation*, John Wiley & Sons, Hoboken N.J.
- Longley, Paul A. 2007, *Geographical information systems and science*, Wiley, Chichester. Online verfügbar unter: <http://www.loc.gov/catdir/enhancements/fy0627/2005298117-d.html> / <http://www.loc.gov/catdir/enhancements/fy0627/2005298117-t.html> [20 Oktober 2009].
- Marr, David 1982, *Vision. A computational investigation into the human representation and processing of visual information*, Freeman, San Francisco.
- Mott, Claudius 2005, *Objektorientierte Klassifikationsstrategien zur Erfassung der Landnutzung aus hochauflösenden Fernerkundungsdaten*. Dissertation., München.
- Mühlenberg, Michael 1997, *Kulturlandschaft als Lebensraum*, Quelle und Meyer, Wiesbaden. Online verfügbar unter: <http://www.bsz-bw.de/cgi-bin/ekz.cgi?SWB05960808> / <http://www.gbv.de/dms/bs/toc/227884825.pdf> [20 Oktober 2009].
- Næset, Erik, Gobakken, Terje, Holmgren, Johan, Hyypä, Hannu, Hyypä, Juha, Maltamo, Matti, Nilsson, Mats, Olsson, Håkan, Persson, Åsa & Ulf Söderman 2004, 'Laser scanning of forest resources: the nordic experience' in *Scandinavian Journal of Forest Research*, pp. 482–499.
- Neuroth, Heike, Kerzel, Martina & Gentsch, Wolfgang 2007, *German Grid Initiative. D-Grid*, Univ.-Verl. Göttingen, Göttingen.

Oczipka, Martin 2007, *Objektbasierte Klassifizierung hochauflösender Daten in urbanen Räumen unter besonderer Berücksichtigung von Oberflächenmodellen*. Dissertation, FU Berlin.

Open Grid Forum 2009, *OGF*. Online verfügbar unter: www.ogf.org [08 September 2009].

Pinz, Axel 1989, 'Final results of the Vision Expert System VES. Finding Trees in Aerial Photographs'. Workshop of the Austrian Association for Pattern Recognition in *Wissensbasierte Mustererkennung (Knowledge-Based Pattern Recognition)*, Oldenbourg, Vienna, pp. 90–111.

Pinz, Axel 1994, *Bildverstehen*, Springer, Wien.

Pitkänen J., Maltamo M., Hyypä J. & Wei Yu X. 2004, 'Adaptive Methods for Individual Tree Detection on Airborne Laser Based Canopy Height Model' in *International Archives of Photogrammetry*, pp. 307–312.

Plaza, Antonio J 2008, *High performance computing in remote sensing*, Chapman & Hall, Boca Raton, Fla. Online verfügbar unter: <http://www.loc.gov/catdir/toc/ecip0718/2007020736.html> [20 Oktober 2009].

Pollock, Richard James 1996, *The automatic recognition of individual trees in aerial images of forests based on a synthetic tree crown image model*, The University of British Columbia (Canada).

Reitberger, J., Krzystek, P. & Stilla, U. 2008, 'Analysis of full waveform LIDAR data for the classification of deciduous and coniferous trees', *Int. J. Remote Sens.*, vol. 29, no. 5, pp. 1407-1431.

Reitberger, J., Schnörr, Cl, Krzystek, P. & Stilla, U. 2009, '3D segmentation of single trees exploiting full waveform LIDAR data' in *ISPRS Journal of Photogrammetry and Remote Sensing*.

Schleyer, A (ed.) 2001, *Das Laserscanning-DGM von Baden-Württemberg*, Wichmann, Heidelberg.

Schmieder, K 2008, *Einsatz von Fernerkundungs- und GIS-Verfahren zur Erfassung von Streuobstwiesenflächen und Baumbestandszahlen in Baden-Württemberg. Forschungsantrag*.

Sixel, Tobias, Straub, Christoph & Wagelaar, Rainer 2005, 'Einsatz von Laserscan-Daten und Orthophotos zur dreidimensionalen Erfassung von Gehölzen im Offenland'. *Angewandte Geoinformatik 2005. Beiträge zum 17. AGIT-Symposium Salzburg*, eds Strobl, Blaschke & Griesebner, Wichmann, Heidelberg.

Sloot, Peter M.A 2005, *Advances in grid computing - EGC 2005. European Grid Conference, Amsterdam, The Netherlands, February 14 - 16, 2005 ; revised selected papers*, Springer, Berlin. Online verfügbar unter: <http://www.loc.gov/catdir/enhancements/fy0663/2005928161-d.html> [20 Oktober 2009].

Soille, Pierre 2003, *Morphological image analysis. Principles and applications*, Springer, Berlin. Online verfügbar unter: <http://www.zentralblatt-math.org/zmath/en/search/?an=1012.68212> [20 Oktober 2009].

Stächele, Willi 2003, *Streuobstbau ist ein Kulturgut Baden-Württembergs. Baden-Württemberg beim Streuobstbau an der Spitze*. Pressemitteilung 20/2003. Online verfügbar unter: http://www.mlr.baden-wuerttemberg.de/content.pl?ARTIKEL_ID=16570 [20 Oktober 2009].

Sterling, Thomas, Becker, Donald J., Savarese, Daniel, Dorband, John E., Ranawake, Udaya A., Packer, Charles V. & Sterling, Thomas 1995, 'Beowulf: A Parallel Workstation For Scientific Computation', *IN PROCEEDINGS OF THE 24TH INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING*, vol. 1, pp. 11–14.

Straub, Bernd-Michael 2003, *Automatische Extraktion von Bäumen aus Fernerkundungsdaten*. Univ., Diss.--Hannover, 2003., Beck, München.

Strellen, Thomas 2003, *Erfassung von Hochspannungsleitungen und Schutzräumen aus Laserscandaten. Entwicklung eines Workflows zur Verarbeitung von dreidimensionalen Punktwolken*. Master Thesis, Salzburg.

Tiede, Dirk, Osberger, Antonia & Novak, Hermann 2009, 'Automatisierte Baumextraktion mit höchst aufgelösten Oberflächenmodellen abgeleitet aus UltracamX-Daten' in *Angewandte Geoinformatik 2009. Beiträge zum 21. AGIT-Symposium Salzburg*, eds J Strobl, T Blaschke & G Griesebner, Wichmann, Heidelberg, p. 55.

TopoSys 2009, *Systemspezifikation LIDAR Systems*. Online verfügbar unter: www.toposys.de.

University of California 2009, *BOINC: Berkeley Open Infrastructure for Network Computing. Open-Source Software für Volunteer Computing und Grid Computing*. Online verfügbar unter: boinc.berkeley.edu [20 October 2009].

Vincent, Luc & Soille, Pierre 1991, 'Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations', *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583-598.

VMware Server 2009, VMware. Online verfügbar unter: <http://www.vmware.com/de/products/server/> [20 October 2009].

Wever, Christian & Lindenberger, Joachim 1999, 'Laser Scanning – A Mapping Method Gains Ground' in *Photogrammetric Week*, pp. 125–132.

Zhang, L, Zhu, J, Zeng, Z & Zhang, L 2005, *A BOINC based System for Global Topographic Structure Extraction Using SRTM Digital Elevation Models*, Beijing.

8.2 Softwareverzeichnis

- ArcGIS 9.2 (<http://www.esri.com>)
- Literaturverwaltungssoftware Citavi (<http://www.citavi.com>)
- Microsoft Office 2007 (<http://office.microsoft.com/de-de/default.aspx>)
- Microsoft Office Visio 2007 (<http://office.microsoft.com/de-de/visio/default.aspx>)
- Matlab Version 7.8 (www.mathworks.com) mit den Erweiterungen
 - Image Processing Toolbox
 - Mapping Toolbox
 - Statistics Toolbox
- BOINC (boinc.berkeley.edu)
- Skriptsprache Python (www.python.org)
- FWTools (GDAL, OGR, fwtools.maptools.org)
- PostGIS (postgis.refractory.net)
- Virtualisierungssystem VMware (www.vmware.com)