

Architektur eines wissensbasierten Systems zur Interpretation multisensorieller Fernerkundungsdaten

Vom Fachbereich Elektrotechnik und Informationstechnik
der Universität Hannover
zur Erlangung des akademischen Grades

Doktor-Ingenieur

genehmigte

Dissertation

von

Dipl.-Ing. Martin Pahl

geboren am 14. März 1968 in Oldenburg(Nds.)

2003

1. Referent: Prof. Dr.-Ing. C.-E. Liedtke

2. Referent: Prof. Dr.-Ing. H. Garbe

Tag der Promotion: 15.04.2003

Martin Pahl

**ARCHITEKTUR EINES WISSENSBASIERTEN
SYSTEMS ZUR INTERPRETATION
MULTISENSORIELLER FERNERKUNDUNGSDATEN**

ibidem-Verlag
Stuttgart

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

∞

Gedruckt auf alterungsbeständigem, säurefreiem Papier
Printed on acid-free paper

ISBN: 3-89821-273-4

© *ibidem*-Verlag
Stuttgart 2003
Alle Rechte vorbehalten

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und elektronische Speicherformen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

Printed in Germany

Wissensbasierte Bildauswertung

Schriftenreihe des Instituts für Theoretische Nachrichtentechnik und Informationsverarbeitung der Universität Hannover

Vorwort des Herausgebers

Die Bildauswertung stellt im Rahmen der Informationsverarbeitung eine junge Disziplin dar. Die Verfügbarkeit neuerer und verbesserter Sensorsysteme und steigende Prozessorleistung für die Auswertung führen zu einem breiten Einsatz von bildverarbeitenden Systemen in fast allen Bereichen der modernen Wirtschaft - von der Kommunikations- und Unterhaltungstechnik über die Medizintechnik, Sicherheitstechnik, Verkehrstechnik, Automatisierungstechnik, Qualitätsprüfung, die Fernerkundung bis zur weltweiten Überwachung der natürlichen Ressourcen.

Aufgaben der Bildauswertung entspringen in der Regel Aufgaben, die in einem vorangegangenen Schritt von einem Menschen mit dessen visuell-kognitiven Fähigkeiten bewältigt worden sind. Die heutigen technischen Bildauswertungssysteme stehen deshalb in direkter Konkurrenz zu den visuellen Fähigkeiten des Menschen und werden an diesen gemessen. Noch sind wir weit davon entfernt, in der Erkennungsleistung und Flexibilität dem Menschen gleichzukommen. Der Vorteil der technischen Systeme liegt in der hohen Reproduzierbarkeit der Ergebnisse, der nicht vorhandenen Ermüdung, hohem Durchsatz und relativ geringen Kosten. Diese Vorteile kann man aber derzeit nur nutzen bei einfachen, weitgehend standardisierten Szenen und festen Arbeitsabläufen. Wenn diese Bedingungen nicht vorliegen, ist der Mensch in seiner Erkennungsleistung der maschinellen Bildauswertung weit überlegen.

Die Überlegenheit des menschlichen visuellen Systems ist wesentlich darin begründet, dass der Mensch für die Bildauswertung systematisch Wissen einsetzt. Er ist in der Lage, dieses Wissen scheinbar mühelos zu erlernen und es den jeweiligen Anforderungen entsprechend gezielt einzusetzen. Ziel der aktuellen Forschung ist es, mit technischen Systemen zu vergleichbarer Leistung zu kommen. Ein Schwerpunkt der Arbeiten zur Bildauswertung am Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung bestand und besteht darin, geeignete Verfahren der Wissensrepräsentation, der Wissensnutzung und des Wissenserwerbs zum Einsatz in bildverarbeitenden Systemen zu erarbeiten. Hierbei spielen Verfahren der expliziten Wissensrepräsentation, die dann zu einer sogenannten »wissensbasierten Bildauswertung« führen, eine besondere Rolle. Die vorliegende Schriftenreihe soll einen Beitrag zur Behandlung dieser Thematik liefern, die eine Schlüsselrolle für künftige Automatisierungsaufgaben beinhaltet.

Hannover, Juli 2003

Prof. Dr.-Ing. Claus-E. Liedtke

Kurzfassung

Die vorliegende Arbeit behandelt den Entwurf einer Architektur eines wissensbasierten Systems zur Interpretation multisensorieller Fernerkundungsdaten. Das System steuert den Interpretationsablauf von Luft- und Satellitenbildern und generiert eine symbolische Szenenbeschreibung, die den Objekten im Bild eine Semantik zuordnet. Die symbolische Szenenbeschreibung wird in Form eines Szenenbaumes angelegt. Jeder Baumknoten korrespondiert dabei mit einem Objekt im Datenmaterial. Die genaue Lage jedes Objekts wird mit einem objektspezifischen Identifikationswert in Labelbildern verzeichnet. Mit Hilfe des Baumes und der Labelbilder lassen sich Karten unterschiedlichen Detaillierungsgrades erzeugen.

Um den Objekten in der Szene eine Bedeutung zuweisen zu können, benötigt das System Modellwissen über die zu erwartenden Objekte. Dieses Modellwissen wird in Form einer Objektklassenhierarchie bereitgestellt, die alle in einer Szene zu erwartenden Objektklassen enthält. Jeder Objektklasse kann ein Bildverarbeitungsoperator zugeordnet werden, der in der Lage ist, Suchbereiche für Objekte dieser Klasse in Fernerkundungsdaten zu bestimmen. Ein weiterer der Klasse zugeordneter Operator dient dazu, Objekte vom Typ der jeweils untergeordneten Klassen zu bewerten und zu gruppieren.

Die Analysestrategie orientiert sich an dem hierarchischen Modellwissen. Zunächst wird die in den Fernerkundungsdaten dargestellte Szene mittels objektklassenspezifischer Bildverarbeitungsoperatoren in Suchbereiche aufgeteilt. Für jeden Suchbereich wird ein hypothetisches Objekt vom Typ der die Suche anstoßenden Klasse generiert. Dieser Vorgang wird rekursiv bis zum Erreichen der Hierarchiebasis wiederholt. Auf diese Weise entsteht eine hypothetische Objekthierarchie, auf der in der Konsolidierungsphase gearbeitet wird. Beginnend an der Hierarchiebasis werden die hypothetischen Objekte von dem klassenspezifischen Operator der übergeordneten Objektklasse bewertet und zu neuen Objekten dieser übergeordneten Objektklasse gruppiert. Mit Erreichen der Hierarchiespitze ist eine Objekthierarchie entstanden, die das Modellwissen auf die in den Fernerkundungsdaten dargestellte Szene angewendet widerspiegelt.

Beim Entwurf der hier vorgestellten Architektur wurde besonderer Wert auf eine

einfache und flexible Schnittstelle zur Anbindung externer Datenverarbeitungsprogramme gelegt. Diese werden sowohl zur Segmentierung und Klassifizierung des Bildmaterials als auch zur Bewertung und Gruppierung der Hypothesen eingesetzt. Durch die strikte und in dieser Form neuartige Trennung der problemabhängigen Funktionen und des problemabhängigen Wissens von der problemunabhängigen Analysestrategie, ist das System für den Einsatz in bestehenden Projekten prädestiniert, in denen Bildsegmentierungsroutinen für unterschiedliche Objektklassen bereits vorhanden sind, denen aber die übergeordnete Analysestrategie fehlt.

Die Leistungsfähigkeit der hier entwickelten Architektur wird anhand zweier Beispiele im Bereich der Landnutzungsanalyse gezeigt. Im ersten Beispiel wird ATKIS mit Hilfe von Luftbildern verifiziert. Das zweite Beispiel demonstriert anhand von Satellitenbildern den Einsatz im Umweltmonitoring.

Stichwörter: Fernerkundung, Bildinterpretation, GIS, wissensbasierte Interpretation, hierarchische Analysestrategie

Abstract

The thesis describes the design of a novel system architecture for knowledge based interpretation of multi-sensory remote sensing data. The system proposed takes control of the application flow for the automated interpretation of aerial and satellite images which results in a tree structured symbolic description of the respective scene. Each object detected in the scene corresponds to a node of the tree. The position of an object is represented by a unique identification value in a set of label images. Based on the scene tree and the label images, maps with different level of detail can easily be obtained.

The automated recognition of an object of a particular class requires a priori knowledge about all classes of objects that are expected to be present or likely to occur in the scene. This knowledge is represented by an object class hierarchy which allows two operators to be assigned to each object class. Firstly, region of interest operators, i.e. image operators that determine one or more search areas for the respective object class in remote sensing data, and, secondly operators that are applied for evaluation and grouping of child classes.

The analysis strategy follows the hierarchical structure of the knowledge representation. In a first step, a scene presented to the system is subdivided according to the search areas obtained from the region of interest operators of all classes in the object class hierarchy. Subsequently, for each class, hypothetical instances of the respective class are generated for each search area obtained by its region of interest operator. This procedure is repeated recursively from the top to the base of the hierarchy, yielding a hypothetical hierarchy of object instances in the scene. In a second step, starting from the base of this hierarchy, all hypothetical instances are evaluated and subsequently grouped into new instances of the parent class applying the evaluation and grouping operator of the respective parent class. Again, this is repeated recursively until the top of the hierarchy is reached. The hierarchy created during the latter step finally represents the model knowledge applied to the remote sensing data of the scene.

During the development of the architecture presented, a major focus was put on a simple and flexible interface for the integration of external data processing modules. These can be used for segmentation and classification of images as

well as for evaluation and grouping of hypotheses. Due to the strict separation of problem-dependent functionalities and problem-dependent knowledge on the one hand and the problem-independent analysis strategy on the other hand the system is well suited and ready to be integrated on top of ongoing projects without requiring a redesign of operators in use.

The usability and efficiency of the proposed architecture has been demonstrated by applying it to two examples in the field of the land use analysis. In the first example ATKIS a German GIS is verified by the interpretation of aerial images. The second example demonstrates the use of the system in environmental monitoring by satellite images.

Keywords: remote sensing, automatic image interpretation, GIS, knowledge-based interpretation, hierarchical analysis strategy

Vorwort

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung der Universität Hannover.

Prof. Dr.-Ing. C.-E. Liedtke gilt mein besonderer Dank für die hervorragenden Arbeitsmöglichkeiten, die angenehme Arbeitsatmosphäre sowie die Übernahme des Hauptreferats. Mit seinen vielfältigen Anregungen zum Thema sowie motivierenden Diskussionen hat er das Entstehen dieser Arbeit maßgeblich gefördert und wissenschaftlich begleitet.

Herrn Prof. Dr.-Ing. H. Garbe danke ich für die Übernahme des Koreferats.

Insbesondere haben aber auch die gute Arbeitsatmosphäre und die netten Kollegen mit ihren kritischen Diskussionen, ihrer Hilfsbereitschaft und ihrem fachlichen Rat zum Gelingen dieser Arbeit beigetragen. An erster Stelle möchte ich hier meine beiden Kollegen Jürgen Bückner und Oliver Stahlhut nennen und ihnen für die hervorragende Zusammenarbeit im Team danken. Aber auch meine Kollegen Jochen Wingbermhühle, Stefan Growe, Markus Gerke und Sönke Müller waren zu passenden Hilfestellungen jederzeit bereit.

Bei der Firma TopoSys möchte ich mich für die Bereitstellung der sehr guten multisensoriellen Daten bedanken.

Für eine perfekte Infrastruktur sorgten in dieser Zeit Oliver Stahlhut, Niko Meine und Matthias Schuh u.v.a. ohne dass sie dabei immer persönlich in Erscheinung traten.

Schließlich gilt mein Dank noch all denen, die in der heißen Phase der schriftlichen Ausarbeitung für Korrekturlesen und Beratung bereit standen. Besonders hervorgetan haben sich da meine Kollegen Daniel Schaper, Torsten Wiebesiek, Jürgen Bückner sowie Jochen Wingbermhühle. Aber auch meine Eltern Elisabeth und Manfred Pahl sowie meine Schwester Ulrike Pahl haben hier größtmöglichen Einsatz gezeigt.

... und dann ist da noch meine Familie, die zeitweise etwas zurückstecken musste. Daher nochmal ein Dankeschön an meine Frau Johanna, die in dieser Zeit

mit erhöhtem Einsatz unsere drei Kinder Magdalena (4), Jonathan (2) und Tobias (0) betreuen musste. Und wie Kinder so sind, haben sie mich auch an das Wesentliche erinnert: O-ton Jonathan: Wir müssen mal wieder in den Zoo ...

Hannover, im April 2003

Eine elektronische Version dieser Arbeit, welche die Bilder in Farbe enthält, kann über die Seite <http://www.tnt.uni-hannover.de/~pahl/diss.html> erreicht werden.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Stand der Technik	3
1.2	Ziele und Gliederung der Arbeit	6
2	Grundlagen der Bildinterpretation	9
2.1	Beschreibung der möglichen Fernerkundungsdaten	9
2.1.1	Prozess der Bilddigitalisierung	10
2.1.2	Generierung von digitalen Höhenmodellen	13
2.1.3	Gewinnung von Laserscan-Daten	14
2.2	Verfahren zur Segmentierung	18
2.2.1	Cluster-Analyse im Merkmalsraum	18
2.2.2	Region Growing	19
2.2.3	Split and Merge	20
2.2.4	Color Structure Code	20
2.3	Wissensrepräsentationsformen	22
2.3.1	Wissensbasierte Systeme	23
3	Konzept eines effizienten wissensbasierten Szenenanalysesystems	25
3.1	Eigenschaften bestehender Szenenanalysesysteme	26
3.1.1	VISIONS	26
3.1.2	ERNEST	28
3.1.3	AIDA	30
3.2	Konzept des neuen Szenenanalysesystems GEOAIDA	33

4	Architektur des Szeneninterpretationssystems GeoAIDA	39
4.1	Basis-Bibliotheken für die Verarbeitung von Fernerkundungsdaten	39
4.1.1	Datenformat zum Ablegen von Fernerkundungsdaten . . .	41
4.1.2	Bildklasse für die Manipulation von Fernerkundungsdaten	42
4.1.3	Geo- und Bildkoordinaten	45
4.1.4	Bibliothek zur Verarbeitung von Objektmengen	48
4.1.5	Parser für die Interpretation von XML-Dateien	49
4.2	Funktionsweise der Datenanalyse	51
4.2.1	Datenbasis	52
4.2.2	Semantisches Netz	53
4.2.3	Top-Down-Operatoren	55
4.2.4	Bottom-Up-Operatoren	56
4.2.5	Interaktive Karte	58
4.2.6	Steuerung der Analyse	58
4.3	Ablauf einer Szeneninterpretation	58
4.3.1	Analysebeispiel	61
4.4	Technische Realisierung	66
4.4.1	Taskmanager	67
4.4.2	Analysesteuerung	67
4.4.3	Informationsaustausch mit externen Komponenten . . .	72
4.4.3.1	Bilddatenbeschreibung	73
4.4.3.2	Regionenbeschreibung	74
4.4.3.3	Knotenliste	75
4.4.4	Beschreibung der externen Operatoren	77
4.5	Ergebnisvisualisierung	81
4.5.1	Datenhaltung	81
4.5.2	Interaktive Karte	82
4.6	Anforderungen an die externen Operatoren	86
4.6.1	Bildinterpretationsoperatoren	87
4.6.2	Bewertungs- und Gruppierungsoperatoren	89
4.7	Experimentalsystem zum Operatorentwurf	92
5	Anwendungsbeispiele	95
5.1	Spezielle Operatoren	95
5.1.1	Gebäudedetektion in Laserscan-Daten	96

5.1.2	Texturbasierte Klassifizierung	98
5.1.3	Klassifizierung über Varianz	101
5.1.4	Detektion von Straßen	101
5.1.5	Ein generischer Bewertungsoperator	103
5.2	Beispiele zur luftbildgestützten Landnutzungsanalyse	104
5.2.1	Hierarch. Analyseablauf und Bewertung von Alternativen	106
5.2.2	Vergleich eines Interpretationsergebnisses mit ATKIS . .	109
5.3	Beispiel zur satellitengestützten Landnutzungsanalyse	111
6	Bewertung und Ausblick	115
6.1	Multihierarchieanalyse	115
6.2	Automatische Operatorenauswahl	116
6.3	Quantitative Qualitätsbewertung	117
6.4	Auswertung multitemporaler Datenbestände	117
7	Zusammenfassung	121
	Glossar	125
	Literaturverzeichnis	135

Formelzeichen

Digitalisierung von Bildmaterial

- Φ_r - Strahlungsfluss eines Rasterpunktes
- $\Phi_{r,min}$ - minimaler Strahlungsfluss eines Rasterpunktes im Bild
- $\Phi_{r,max}$ - maximaler Strahlungsfluss eines Rasterpunktes im Bild
- Φ_b - Quantisierter Wert eines Rasterpunktes
- $D(x, y)$ - Intensität (Strahlungsflussdichte) im Punkt (x,y)

Pixel Boundingbox

- x_{pll} - Lower Left x-Wert
- y_{pll} - Lower Left y-Wert
- x_{pur} - Upper Right x-Wert
- y_{pur} - Upper Right y-Wert
- x_p - x-Wert in Pixelkoordinaten
- y_p - y-Wert in Pixelkoordinaten
- x_{size} - Größe eines Bildes in x-Richtung
- y_{size} - Größe eines Bildes in y-Richtung

Geokoordinaten Boundingbox

- x_g - x-Wert in Geokoordinaten
- y_g - y-Wert in Geokoordinaten
- x_{gW} - Geo West
- y_{gN} - Geo North
- x_{gE} - Geo East
- y_{gS} - Geo South
- x_{gll} - Lower Left x-Wert in Geokoordinate
- y_{gll} - Lower Left y-Wert in Geokoordinate

x_{gur} - Upper Right x-Wert in Geokoordinate

y_{gur} - Upper Right y-Wert in Geokoordinate

Semantisches Netz

N - Knoten eines Netzes T_N im allgemeinen

N' - Unterknoten eines Knotens N

N'' - Unterknoten eines Knotens N'

S - Spezieller Knoten N eines Semantischen Netzes T_S

H - Spezieller Knoten N eines hypothetischen Instanzennetzes T_H

I - Spezieller Knoten N eines Instanzennetzes T_I

T_N - Netz (Baum) bestehend aus Knoten N

T_S - Netz (Baum) bestehend aus Knoten S

T_H - Netz (Baum) bestehend aus Knoten H

T_I - Netz (Baum) bestehend aus Knoten I

P - Teilbild, das einem Knoten N zugeordnet ist

P' - Ausschnitt eines Bildes P

T_{P_I} - Baum

Suchkosten im A*-Algorithmus

c_{total} - Geschätzte Gesamtkosten bis zum Erreichen einer Lösung

c_{accum} - Akkumulierte Kosten der bisherigen Suche

$c_{element}$ - Kosten für die Verfolgung eines Teilstücks

c_{remain} - Geschätzte Restkosten bis zum Erreichen einer Lösung

w - Bereits zurückgelegte Wegstrecke in Anzahl Iterationsschritten

s - Maximale Suchtiefe in Anzahl Iterationen

λ - Gewichtungsfaktor

Texturbasierte Klassifizierung

κ - Clique

V_κ - Potential einer Clique

P_1 - Wahrscheinlichkeit für das Auftreten einer bestimmten Luminanz

P_2 - Wahrscheinlichkeit für jede Clique, dass sie derselben Klasse angehören

P_3 - Wahrscheinlichkeit für die Luminanzdifferenz jeder Clique innerhalb einer Region

P_4 - Wahrscheinlichkeit für die Luminanzdifferenz jeder Clique an der Regionengrenze

w_i - Gewichtung für die Potentiale P_i

1 Einleitung

Fernerkundungsdaten werden für die vielfältigsten Anwendungsgebiete ausgewertet. Anfangs diente derartige Datenmaterial überwiegend zur Unterstützung bei der Erstellung von Karten. Heutzutage rücken Aufgaben des Monitoring in Bereichen wie Umweltschutz[58][94], Katastrophenschutz, Landschafts- und Städteentwicklung[3] sowie der Luftaufklärung in Krisengebieten in den Vordergrund. Hinzu kommen Geoinformationssysteme (GIS)[53], wie sie z.B. für die Routenplanung und Navigationshilfen benötigt werden, die auf hohe Aktualität ihrer Daten angewiesen sind.

Treibende Kräfte dieser Entwicklung sind durch vom Menschen verursachte Veränderungen der Erdoberfläche, die eine regelmäßige Datenerfassung für Zukunftsprognosen erforderlich machen. Eine kontinuierliche Datenerfassung wird jedoch erst durch den Einsatz hochauflösender Sensoren, insbesondere in Kombination mit Satellitentechnik, ermöglicht.

Die Erfassung von Fernerkundungsdaten erfolgt sowohl durch Bildflüge als auch mit Hilfe von Satelliten. Dabei kommen Sensoren zum Einsatz, die in unterschiedlichen Spektralbereichen arbeiten, um verschiedene Aspekte einer Szene zu erfassen. Immer höher auflösende Sensoren und eine wachsende Zahl von durchgeführten Bildflügen sowie die kontinuierliche Datenerfassung durch satellitengestützte Systeme generieren ein stark anwachsendes Datenvolumen. Eine rein manuelle Auswertung ist folglich sehr aufwendig, damit unwirtschaftlich und nicht praktikabel. Seit längerem wird deshalb schon an der automatischen Extraktion bestimmter Merkmale aus den Fernerkundungsdaten gearbeitet, um so den Datenanalysten zu unterstützen bzw. dessen Aufmerksamkeit auf bestimmte Bildinhalte zu lenken. Von besonderem Interesse ist dabei neben einer Landnutzungsklassifikation die Detektion künstlicher Objekte wie Straßen[15][102] oder Gebäude[32][36][45][51][63][37]. Zudem gibt es Bestre-

bungen den gesamten Interpretationsvorgang zu automatisieren. Hierzu ist es notwendig, das Vorwissen des menschlichen Analysten, das dieser für eine kontextabhängige Interpretation der Fernerkundungsdaten einsetzt, nachzubilden. Um sich flexibel den unterschiedlichen Inhalten und Anforderungen anpassen zu können, wird dieses Vorwissen in einer externen Wissensbasis vorgehalten. Diese enthält Informationen über in der zu betrachtenden Szene möglicherweise vorhandene Objekte und deren Relationen zueinander.

Bevor der Analyst mit der eigentlichen Auswertung beginnt, wird er sich anhand von zusätzlich verfügbaren Informationen wie Aufnahmedatum und -ort einen groben Überblick über die Szene verschaffen. Bei der Betrachtung der Szene wird er zunächst übergeordnete Objekte erkennen, die sich in immer feinere Teilobjekte zerlegen lassen, bis er schließlich bedingt durch die endliche Auflösung des Datenmaterials die kleinstmöglichen Strukturen erfassen wird[39]. Manche Objekte lassen sich dabei erst durch Betrachten ihrer Teilobjekte genau identifizieren, während andere in einem größeren Zusammenhang leichter erkennbar sind. Zu jedem Zeitpunkt ist es möglich, verfügbare Informationen z.B. aus Karten oder einem GIS in den Interpretationsprozess einzubringen. Diese Informationen lenken die Aufmerksamkeit in eine bestimmte Richtung und stellen damit einen Kontext für die weitere Interpretation her.

Die vorliegende Arbeit hat den Entwurf einer effizienten Architektur zur automatischen Interpretation von Fernerkundungsdaten in Anlehnung an das Vorgehen eines menschlichen Analysten zum Thema. Eine hierarchische Wissensbasis, welche die Multiskaligkeit der in den Fernerkundungsdaten vorhandenen Objekte widerspiegelt, soll dabei zur Steuerung der Analyse eingesetzt werden. So wie der menschliche Betrachter jederzeit verfügbare Zusatzinformationen in den Interpretationsprozess einfließen lassen kann, soll auch die automatische Interpretation in der Lage sein, durch zusätzliche Informationen die Interpretation zu lenken. Für die Detektion spezieller Objektklassen in den Luft- und Satellitenbildern sollen bereits vorhandene Verfahren eingesetzt werden. Die Integration dieser Bildinterpretationsprogramme soll über eine flexibel zu konfigurierende Schnittstelle erfolgen. Für die strukturelle Analyse, die Teilobjekte zu übergeordneten Objekten gruppiert, ist ebenfalls eine Schnittstelle zur Anbindung externer Bewertungsprogramme vorzusehen, so dass die Modellierung spezieller Objektrelationen, die zum Zeitpunkt des Erstellens dieser Arbeit

nicht absehbar sind, ermöglicht wird. Durch dieses offene Konzept soll die zu entwickelnde Architektur für den nachträglichen bzw. begleitenden Einsatz in bestehenden Projekten prädestiniert sein, um dort übergeordnete, wissensbasierte Komponenten einzubringen.

1.1 Stand der Technik

Wissensbasierte Bildinterpretation wird mit der Zielsetzung betrieben, Objekte im Bild zu erkennen und ihnen eine Bedeutung zuzuweisen. Die Bildinterpretation gliedert sich in zwei Bereiche, die Bildsegmentierung, die zusammenhängende Regionen aus einem Bild extrahiert, und die symbolische Verarbeitung, die den extrahierten Regionen auf Grund bestimmter Merkmale eine initiale Bedeutung zuweist sowie unter Berücksichtigung von Objektrelationen übergeordnete Objekte generieren kann.

Die ersten Systeme zur wissensbasierten Luftbildinterpretation stammen aus den achtziger Jahren. Zu ihnen gehören ACRONYM[21], BPI[70], ECHO[55], MESSIE[30], SIGMA[71][72], SPAM[73] und VISIONS[47]. Sie wurden in erster Linie zum Erkennen von Einzelobjekten wie Gebäuden, Straßen oder Flugzeugen eingesetzt. Zur Steuerung der Interpretation verwenden die meisten dieser Systeme Regeln, die in Regelbasen vorgehalten werden. Der Umfang dieser Regelbasen steigt mit der Komplexität der Szenen schnell an, wodurch sie unübersichtlich werden. Dies gilt insbesondere für Systeme wie SPAM und BPI, die auch das gesamte Modellwissen in Regeln formulieren. Die Systeme SIGMA und VISIONS beschreiben ihr Modellwissen hingegen mit Hilfe semantischer Netze.

Ein semantisches Netz wird ebenfalls im System ERNEST[80] zur expliziten Wissensrepräsentation eingesetzt. Dieses Vorwissen wird mit wenigen anwendungsunabhängigen Regeln ausgewertet. Die Regeln sind dabei jedoch impliziter Bestandteil des Steueralgorithmus. Diesen Schwachpunkt greift das System AIDA[67] auf, indem es auch die Steuerungsregeln explizit formuliert.

Zur Bewältigung von Konfliktsituationen und Mehrdeutigkeiten verwenden sowohl ERNEST als auch AIDA eine heuristische Suche und verfolgen immer

nur die vielversprechendste Alternative, wofür ein Bewertungssystem erforderlich ist. Die Systeme ACRONYM, SPAM und SIGMA hingegen berücksichtigen zunächst alle möglichen Alternativen und treffen erst zum Schluss eine Wahl. Dies ist jedoch angesichts der komplexen Szenen in der Fernerkundung eine unbrauchbare Methode. In [42][83] wurde das System AIDA zur multitemporalen Auswertung erweitert.

Neuere Ansätze [19][7][8][74] sind auch in die kommerziellen Produkte *eCognition* der Firma Definiens und den *ExpertClassifier* der Firma ERDAS[35] eingeflossen.

Der *ExpertClassifier* der Firma ERDAS stellt einen regelbasierten Ansatz zur Klassifikation multispektraler Eingangsbilder sowie zur nachträglichen Verbesserung einer initialen Klassifikation dar. Im Kern besteht die Klassifikation aus einer Hierarchie von Regeln, die einen Entscheidungsbaum darstellt. Dieser Entscheidungsbaum beschreibt die Bedingungen, unter denen ein Satz von Informationen in eine höhere Abstraktionsebene gewandelt wird. Die Eingangsinformationen bestehen dabei aus benutzerdefinierten Variablen, die beispielsweise aus Bilddaten direkt oder indirekt abgeleitet sein können. Es kann sich dabei aber auch um Vektordaten z.B. aus einem GIS handeln.

Eine Regel ist eine bedingte Anweisung oder eine Liste bedingter Anweisungen, die angewandt auf die benutzerdefinierten Variablen zum Aufstellen einer Hypothese führt. Mehrere Regeln und Hypothesen können zu einer Hierarchie verbunden werden, die den Weg der Eingangsdaten zu einem finalen Satz von Klassen bzw. Hypothesen weist. Mit jeder Regel können Vertrauensmaße verknüpft werden, die eine Bewertung der Ergebnisklassen zulassen. Dieser rein pixelbasierte Ansatz bezieht jedoch Nachbarschaftsbeziehungen nicht in die Analyse mit ein.

Die grundlegende Idee von *eCognition* ist, dass wichtige semantische Informationen nicht in Pixeln, sondern in Objekten und deren Beziehungen zueinander zu finden sind[6]. Mittels einer Segmentierung auf Basis der „Fractal Net Evolution“-Technik, welche die Segmente aufgrund von Homogenitätskriterien[18] bildet, wird ein hierarchisches Netzwerk von Bildobjekten aufgebaut. Auf diese Weise sind jedem Bildobjekt seine Nachbarn bekannt und es können Beziehungen zwischen diesen definiert werden. Dies geschieht mit Hilfe einer Klassen-

hierarchie, die als semantisches Netz bezeichnet werden kann. Die eigentliche Klassifizierung findet dabei entweder durch Zuordnung einer zuvor angelernen Region oder mittels einer vorzugebenen Berechnungsvorschrift statt. Dabei können Relationen zu benachbarten, über- und untergeordneten Knoten eingebracht werden.

Nachteilig ist bei diesem Ansatz, dass das bisher nicht veröffentlichte Segmentierungsverfahren fest vorgegeben ist. Zudem müssen alle Eingangsdaten über die gleiche Auflösung verfügen.

Eine zuverlässige Klassifikation kann nur durch eine gute Regelbasis erfolgen. In [13] wird daher das Structural Analysing and Mapping System (SAMS)[11] eingesetzt, um mittels eines klassifizierten Bildausschnitts, Attribute der Objekte und Objektrelationen für den Aufbau des Regelwerks in *eCognition* zu ermitteln.

Sowohl *eCognition* als auch *ExpertClassifier* sind als kommerzielle Systeme auf Kundenbindung ausgerichtet und erlauben als solche nur eine eingeschränkte Integration von Fremdprodukten. Hat man sich einmal für ein System entschieden, so ist man auf die vom jeweiligen Herstellers implementierten Fortschritte angewiesen.

Keines der genannten Systeme ist bisher für eine vollautomatisierte Auswertung von Fernerkundungsdaten geeignet. Während sich die kommerziellen Systeme auf rein pixelorientierte Ansätze konzentrieren, die keine Nachbarschaftsbeziehungen berücksichtigen, bzw. in ihren Bildverarbeitungsoperatoren bezüglich der Segmentierung festgelegt sind, hat beispielsweise das wissensbasierte System AIDA den Nachteil, dass es nach einer initialen Extraktion von Bildprimitiven rein strukturell arbeitet. Um der Komplexität von Fernerkundungsdaten gerecht zu werden, ist aber ein hierarchischer, multiskaliger Ansatz erforderlich. Dies bedeutet, dass auf Grundlage einer Objekthierarchie, die als Modellwissen eingebracht wird, frühzeitig versucht wird, den Suchraum für übergeordnete Objekte einzuschränken. Hierzu wird jeweils der im betreffenden Kontext optimale Operator eingesetzt. Umgekehrt wird Wissen über die Objektattribute und -relationen eingesetzt, um in den Suchbereichen Objekte einer bestimmten Klasse zu identifizieren. Diese effiziente Vorgehensweise bei gleichzeitiger Integrationsfähigkeit für bestehende Algorithmen wird bisher von keinem der

genannten Systeme umgesetzt.

1.2 Ziele und Gliederung der Arbeit

Ziel dieser Arbeit ist es, eine effiziente Architektur zur wissensbasierten Interpretation von Fernerkundungsdaten zu entwickeln. Ein wichtiger Aspekt soll dabei die Integrierbarkeit bestehender Bildverarbeitungsprogramme zur Detektion spezieller Objektklassen sein. Das Modellwissen soll in einem semantischen Netz abgelegt werden, das mit seinem hierarchischen Aufbau die in den Fernerkundungsdaten erwartete Objekthierarchie widerspiegelt und gleichzeitig zur Steuerung der Analyse dient.

Bisherige Verfahren gehen von Bildprimitiven aus und gruppieren diese zu übergeordneten Objekten, wobei die Aufgabe darin besteht, die beste Gruppierungskombination in Bezug auf das Modellwissen zu finden. Im Gegensatz dazu wird in dieser Arbeit angestrebt, den Suchbereich für mögliche Objekte mit der Hierarchiespitze des Modellwissens beginnend durch den Einsatz sogenannter holistischer Operatoren, die ein Bild in der für die Detektion der gesuchten Objekte günstigsten Skalierung ganzheitlich bearbeiten, einzuschränken, um dann in untergeordneten Hierarchieebenen nur noch kontextspezifisches Wissen einbringen zu müssen. Durch diese kontextabhängige Einschränkung des Suchraums wird die Zahl der Kombinationsmöglichkeiten auf der Ebene der Bildprimitive stark begrenzt. Diese hypothetische Objekthierarchie soll anschließend mit den Blattknoten beginnend verifiziert werden. Dabei sollen auftretende Konflikte möglichst lokal in jedem Teilbaum durch externe Bewertungsoperatoren gelöst werden.

Ein weiterer Teil der Arbeit beschäftigt sich mit der Frage, wie die gewonnenen Ergebnisse einem potentiellen Nutzer übersichtlich dargestellt werden können. Hierzu sollen das Modellwissen und die ermittelte Objekthierarchie mit einer kartenähnlichen Darstellung interaktiv verknüpft werden.

Die vorliegende Arbeit gliedert sich daher wie folgt:

- Kapitel 2 beschreibt das verwendete Bildmaterial und dessen relevante Ei-

enschaften, wie sie für die automatische Extraktion von Objekten wichtig sind. Grundvoraussetzung für eine Bild-Interpretation ist eine Segmentierung. Die grundsätzlichen Methoden zur Segmentierung werden daher dargestellt. Da eine Bedeutungszuweisung nur durch Einbringen von Vorwissen möglich ist, werden einige wichtige Wissensrepräsentationsformen vorgestellt.

- In Kapitel 3 wird das Konzept des im Rahmen dieser Arbeit entstandenen wissensbasierten Szenenanalysesystems GEOAIDA im Vergleich zu bereits bestehenden Systemen dargestellt.
- Funktionsweise und Implementierungsdetails des Szenenanalysesystems GEOAIDA werden im Kapitel 4 behandelt.
- In Kapitel 5 werden einige im Rahmen des Projekts entstandene Operatoren zur Detektion spezieller Objektklassen in den Fernerkundungsdaten vorgestellt sowie anhand zweier Anwendungsbeispiele der erfolgreiche Einsatz des Systems demonstriert.
- In Kapitel 6 werden, nach einer kurzen Bewertung der im Rahmen dieser Arbeit entwickelten Architektur, mögliche Entwicklungsrichtungen zukünftiger Arbeiten aufgezeigt.
- Kapitel 7 fasst den Inhalt dieser Arbeit zusammen.
- Erklärungen zu Fachbegriffen sowie Abkürzungen sind in einem Glossar am Ende dieser Arbeit zu finden.

2 Grundlagen der Bildinterpretation

Die Interpretation von Fernerkundungsdaten beinhaltet eine Kette von Arbeitsschritten. Am Anfang steht die Akquisition des Datenmaterials. Dabei kommen unterschiedliche Verfahren zum Einsatz, welche die nachfolgende Verarbeitungskette beeinflussen. In jedem Fall müssen jedoch die Daten in Segmente mit ähnlichen Eigenschaften zerlegt werden, bevor ihnen schließlich mit Hilfe von Vorwissen eine Bedeutung zugewiesen werden kann. In den folgenden Abschnitten soll auf diese Themenkomplexe näher eingegangen werden. Abschnitt 2.1 beschäftigt sich mit wichtigen Eigenschaften der verschiedenen Eingangsdaten. In Abschnitt 2.2 werden einige häufig verwendete Segmentierungsalgorithmen vorgestellt. Abschnitt 2.3 gibt einen Überblick über mögliche Verfahren, Wissen in den Bildinterpretationsprozess einzubringen.

2.1 Beschreibung der möglichen Fernerkundungsdaten

Fernerkundungsdaten lassen sich je nach verwendeter Aufnahmeplattform, Messverfahren und Speichermedium in verschiedene Klassen einteilen. Bei der Aufnahmeplattform ist zwischen satelliten- und flugzeuggestützter zu unterscheiden. Als Messverfahren werden Intensitäts- sowie Laufzeitmessungen durchgeführt. Für die Laufzeitmessungen kommen grundsätzlich nur aktive Sensoren in Frage. Intensitätsmessungen können sowohl aktiv als auch passiv durchgeführt werden. Als Speichermedium wurden früher nur analoge Filme eingesetzt, die sich aber nur für die Archivierung von Intensitätssignalen eignen. Inzwischen setzen sich mehr und mehr digitale Aufzeichnungsformen durch, nicht zuletzt wegen ihrer einfacheren Weiterverarbeitungsmöglichkeiten in Rechenan-

lagen.

Farb-, Grauwert- sowie Infrarotbilder können satelliten- und flugzeuggestützt mit passiven Sensoren erfasst werden. Die Sensoren unterscheiden sich lediglich durch ihre Empfindlichkeit in bestimmten Spektralbereichen des Lichtes. Für die von den Landesvermessungsämtern durchgeführten Bildflüge werden häufig noch Reihenschichtkameras (RMK)[59][96] eingesetzt, die das Luftbild auf einem Film aufzeichnen. Für eine Weiterverarbeitung im Rechner müssen diese Filme digitalisiert werden.

Nachteil dieser passiven Messverfahren ist ihre Wetter- und Tageszeitabhängigkeit. Dies führte zur Entwicklung des Radars mit synthetischer Apertur (SAR)[33][65][95]. Hierbei handelt es sich um einen aktiven Sensor. Die ausgestrahlten Mikrowellen durchdringen Wolken und Niederschlag. Durch Auswertung interferometrischer Eigenschaften der reflektierten Strahlen ist sogar eine Tiefenmessung möglich.

2.1.1 Prozess der Bilddigitalisierung

Um Bildmaterial auf Rechenanlagen verarbeiten zu können, muss dieses in digitaler Form vorliegen. Der Prozess der Bilddigitalisierung, der bei digital aufzeichnenden Sensoren während der Datenaufnahme stattfindet, bei analogen hingegen zu einem späteren Zeitpunkt mittels eines Scanners durchgeführt wird, lässt sich in zwei Schritte untergliedern, die Rasterung des Bildmaterials und die Quantisierung der gerasterten Helligkeitswerte bzw. bei Farbbildern zusätzlich der Farbsättigung und des Farbwertes. Die Qualität der späteren Bildinterpretation hängt entscheidend von der Qualität des Digitalisierungsprozesses ab.

Bei der Rasterung werden die innerhalb eines Rasterpunktes liegenden Strahlungsflussdichten (Intensitäten) $D(x, y)$ zu einem Strahlungsflusswert Φ_r , der im folgenden auch als Helligkeitswert bezeichnet wird, integriert:

$$\Phi_r = \int \int D(x, y) dx dy$$

Dies hat zur Folge, dass reale geometrische Strukturen im digitalen Bild in eine

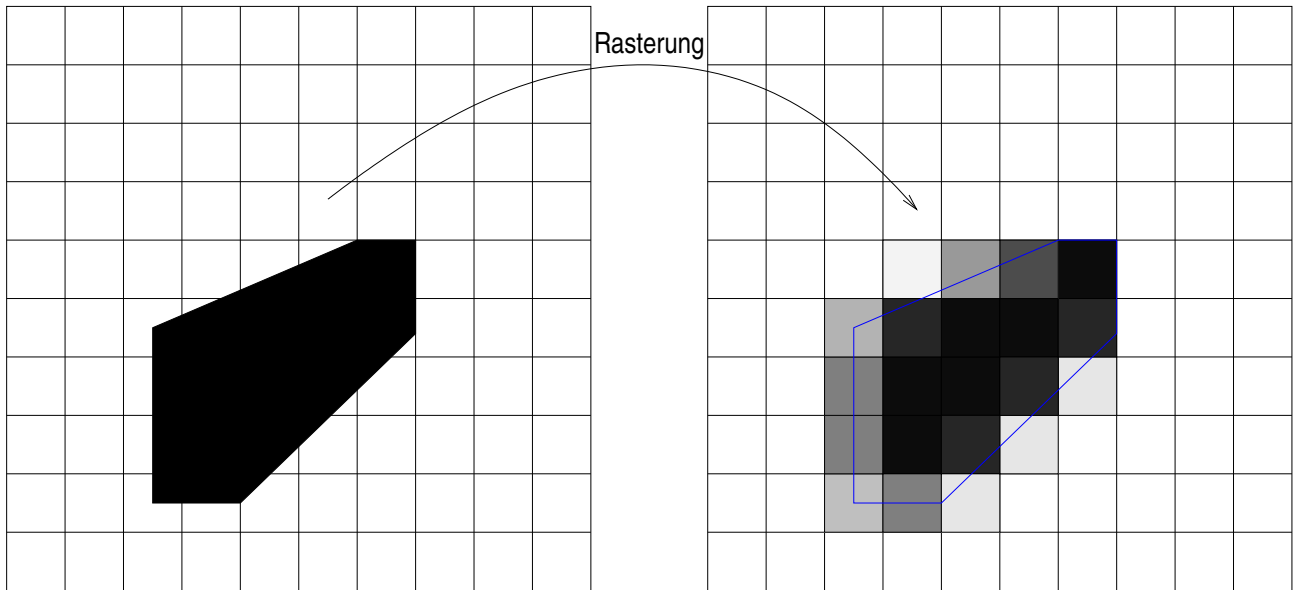


Abbildung 2.1: Die Rasterung von Geometrie führt durch die integrierenden Eigenschaften der lichtempfindlichen Sensoren zu einer Tiefpassfilterung der Kanten.

Kombination aus geometrischer Struktur und Helligkeitsverteilung überführt werden (s. Abbildung 2.1). Dies erschwert die Detektion von Formen, da im Bild nicht mehr erkennbar ist, ob ein Intensitätswert Materialeigenschaft oder Struktur ausdrückt. Es ist zu erwarten, dass jede reale Kante im Bild zu einem mindestens ein Pixel breiten Helligkeitsübergang führt.

Bei der Quantisierung werden die realen, gerasterten Helligkeitswerte Φ_r auf einen quantisierten Helligkeitswert Φ_b , der im Bereich eines im Rechner verfügbaren Wertebereichs liegt, abgebildet. Allgemein geschieht dies über die folgende im Bereich der Helligkeitswerte $[\Phi_{r,min}, \Phi_{r,max}[$ lineare Abbildung:

$$\Phi_b = \begin{cases} 0 & \Phi_r < \Phi_{r,min} \\ \text{int} \left(\frac{\Phi_r - \Phi_{r,min}}{\Phi_{r,max} - \Phi_{r,min}} \cdot 2^n \right) & \Phi_{r,min} \leq \Phi_r < \Phi_{r,max} \\ 2^n - 1 & \Phi_r \geq \Phi_{r,max} \end{cases}$$

Dabei stellt n die maximale Breite eines digitalen Wertes in Bit dar, $\Phi_{r,min}$ ist der real vorhandene Helligkeitswert, der auf den digitalen Wert 0 abgebildet werden soll, i.d.R. also die minimal vorhandene Helligkeit. Entsprechend wird

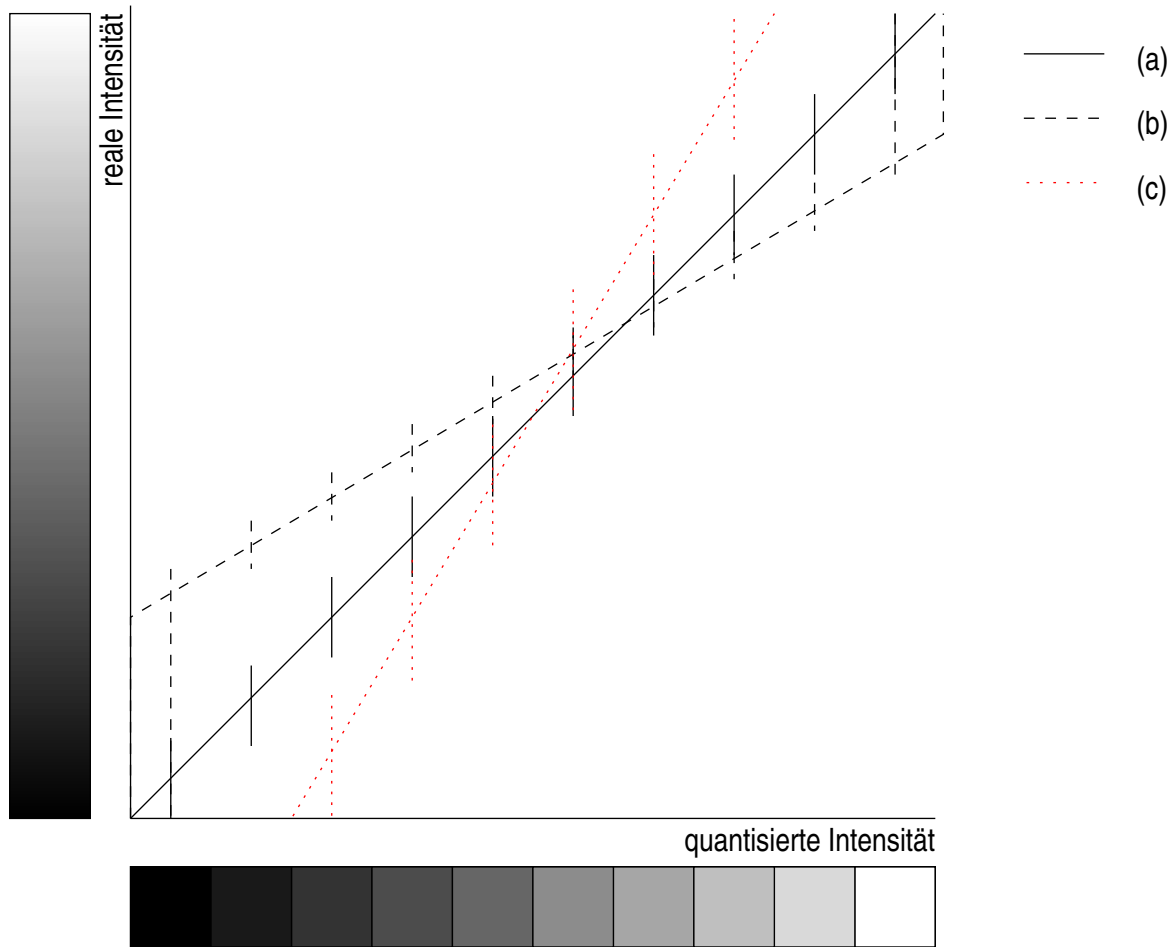


Abbildung 2.2: *Quantisierung der Bildintensität: (a) optimale Abbildung der Intensitätsdynamik, (b) Abbildung eines Teilbereichs der realen Dynamik auf die Bilddynamik verringert den Quantisierungsfehler, (c) schlechte Ausnutzung der Bilddynamik bei gleichzeitiger Vergrößerung des Quantisierungsfehlers.*

$\Phi_{r,max}$, die maximale Helligkeit, auf den digitalen Wert $2^n - 1$ abgebildet.

Abbildung 2.2 veranschaulicht diesen Quantisierungsvorgang. Der Kurvenverlauf (a) beschreibt dabei den bezüglich der vollständigen Helligkeitsdynamik optimalen Quantisierungsprozess. Manchmal kann auch eine Quantisierung nach (b) sinnvoll sein, wenn nur unwichtige Bildinhalte in der Sättigung im Schwarz- bzw. Weißbereich verschwinden, da sich der Quantisierungsfehler im abgebildeten Bereich verkleinert und in dem verstärkten Intensitätsbereich mehr Details sichtbar werden. Eine Quantisierung nach (c) sollte nach Möglichkeit vermie-

den werden, da hier eigentlich darstellbare Information verloren geht. Zudem wird der Quantisierungsfehler unnötigerweise erhöht. Der Fall (c) entsteht, wenn $\Phi_{r,min}$ kleiner als die minimale, im Bild vorhandene Helligkeit gewählt wird, bzw. wenn $\Phi_{r,max}$ größer als die maximale Helligkeit ist. Da der optimale Kurvenverlauf nur bei bekannter minimaler und maximaler Helligkeit erreicht werden kann, ist es i.d.R. sinnvoll mit einer größeren Anzahl von Bits, als tatsächlich benötigt, zu digitalisieren. Dabei ist zu berücksichtigen, dass bei einer Reduktion der Wortbreite ein nichtlinearer Quantisierungsfehler entsteht.

2.1.2 Generierung von digitalen Höhenmodellen

Digitale Höhenmodelle können, soweit sie die Modellierung von Objekten mit einschließen, sehr hilfreich bei der Interpretation von Fernerkundungsdaten sein, da sich aus ihnen Strukturen eindeutig erkennen lassen, wohingegen sich in Luftbilddaten Textur und Struktur vermischen. Bei digitalen Geländemodellen, die aus einer Landvermessung entstanden sind, werden künstliche Objekte bewusst nicht beachtet. Die Erstellung solcher Modelle ist aufwändig und praktisch nicht automatisierbar.

Digitale Geländemodelle aus Stereobildern Besser geeignet im Sinne einer automatischen Generierbarkeit sind aus Stereobildern berechnete Höhenmodelle[4][10][9][64]. Ähnlich wie das räumliche Sehen des Menschen, basieren die Verfahren zur stereoskopischen Bildauswertung darauf, korrespondierende Bildinhalte in einem Stereobildpaar zu finden. Aus der Verschiebung der korrespondierenden Bildinhalte in den beiden Stereobildern, der sogenannten Disparität, lässt sich bei bekannter Kameraanordnung direkt ein Maß für die Tiefe des zum Bildinhalt gehörigen Raumpunktes ableiten. Zwei Klassen von Disparitätsverfahren lassen sich unterscheiden. Die eine Klasse wertet die Intensität des Bildsignals direkt aus und ermöglicht so die Schätzung der Disparität in nahezu jedem Bildpunkt. Dagegen schätzen die merkmalsbasierten Verfahren die Disparität zuvor in beiden Bildern extrahierter Merkmale. Nur für die Merkmalspositionen werden Disparitätswerte bestimmt.

Aus Stereobildern berechnete Höhenmodelle eignen sich für die Berechnung von

Geländemodellen. Für die Modellierung künstlicher Objekte bedarf es Tiefenkarten der Bauwerke mit einem Punktabstand von weniger als $2m$, deren Berechnung sehr aufwändig und mit den in der Photogrammetrie derzeit eingesetzten Verfahren kaum zu leisten ist. Die für die Kommunikationstechnik entwickelten Verfahren hingegen erreichen zwar ein sehr hohe Genauigkeit der Disparitätsschätzung, die teilweise unter der Größe eines Bildpunktes liegt, sind aber mit den in der Fernerkundung anfallenden Bildgrößen überfordert.

2.1.3 Gewinnung von Laserscan-Daten

Die genauesten und dichtesten Höhenmodelle lassen sich derzeit mit einem flugzeuggestützten Laserscanner erreichen. Bei einer Flughöhe von $1000m$ sind Genauigkeiten in der Höhenmessung von weniger als $0,1m$ bei einer Rasterdichte von $1m$ möglich.

Der Laserscanner ist ein aktiver Sensor. Der Sensor sendet einen Lichtstrahl aus und misst die Zeitdauer, die das Licht für den Hin- und Rückweg vom Sensor zum Objekt benötigt (s. Abbildung 2.3). Die Laufzeit des Lichts ist ein direktes Maß für die Entfernung des Sensors zum Objekt. Wiederholt man die Messungen, indem man den Messstrahl mittels eines Ablenkspiegels zeilenweise auf verschiedene Positionen ablenkt und gleichzeitig die Sensorplattform parallel zum Messobjekt und senkrecht zur Abtastzeile bewegt, so entsteht eine Tiefenkarte der betrachteten Szene.

In Abbildung 2.4 ist der Aufbau eines Laserscanners der Firma TopoSys¹ dargestellt. Mittels eines rotierenden, gekippten Spiegels wird der Laserstrahl in ein Glasfaserbündel und von dort zum Boden gelenkt. Zu jedem Glasfaserstrang des Senders korrespondiert einer des Empfängers, der das reflektierte Signal aufnimmt und über den zweiten Ablenkspiegel, der starr mit dem Sendespiegel gekoppelt ist, in den Empfänger umlenkt. Der Abstand der einzelnen Glasfaserstränge bestimmt zusammen mit der Flughöhe den Abstand der Messpunkte auf der Erdoberfläche in Richtung der Scanzeile. In Flugrichtung wird der Abstand der Messpunkte durch die Fluggeschwindigkeit festgelegt.

¹<http://www.toposys.de>

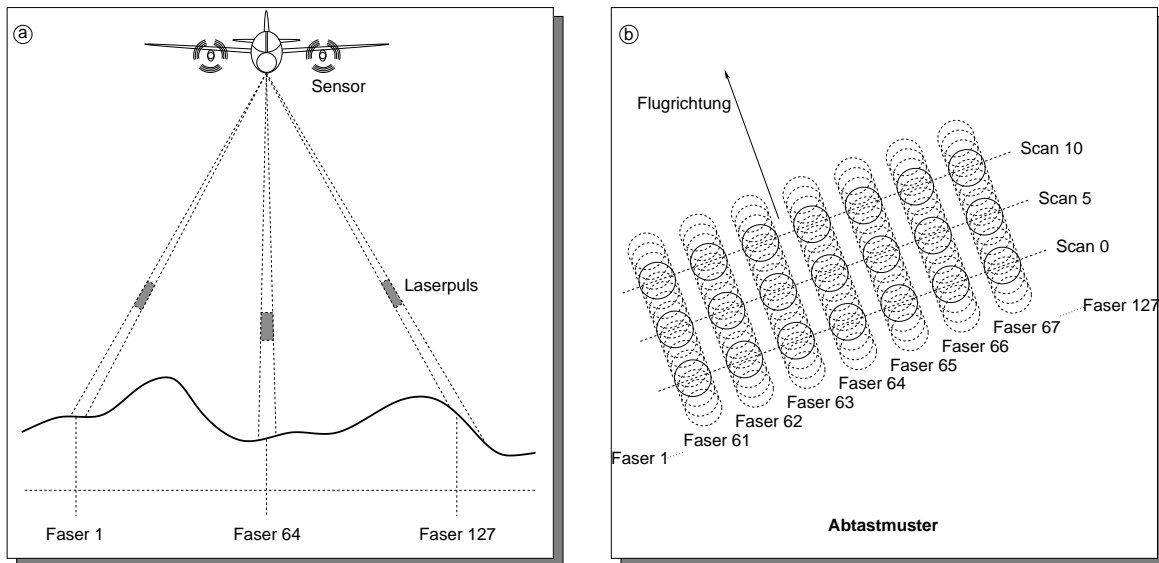


Abbildung 2.3: a) Beim Laserscanning wird die Entfernung vom Flugzeug zum Boden über die Laufzeit des Lichtes gemessen. Aus der Position des Sensors, der Orientierung der Einzelmessung und aus der gemessenen Entfernung lassen sich Ort und Höhe eines Objektes bestimmen. b) Der Laserscanner arbeitet zeilenorientiert, wodurch am Boden ein ebenfalls zeilenförmiges Abtastmuster entsteht. Der Abstand der Faser wird durch den Abstand der Zellen in der Scaneinheit bestimmt. Die Dichte der Zeilen ist abhängig von der Fluggeschwindigkeit und Rotationsgeschwindigkeit des Ablenkspiegels (s. auch Abbildung 2.4).

Da im Gegensatz zu den anderen Messverfahren eine Zeit gemessen wird, entfällt das Integrieren über eine Teilfläche und damit auch die Tiefpassfunktionalität des Integrals. Der Laserscanner bildet Kanten scharf ab. Laserscan-Höhenmodelle eignen sich daher hervorragend als Eingangsdaten für die Detektion künstlicher Objekte. Selbst die Vermessung tiefer Häuserschluchten in dicht bebauten Innenstädten ist möglich, da mit einem Messwinkel von $90^\circ \pm 7^\circ$ gearbeitet wird (s. Abbildung 2.5a).

Beim Auftreffen auf eine Kante ergeben sich mehrere Reflektionen, von denen die erste und letzte aufgezeichnet werden. Mit diesen als First- und Lastpuls bezeichneten Messungen lassen sich beispielsweise halb lichtdurchlässige Vege-

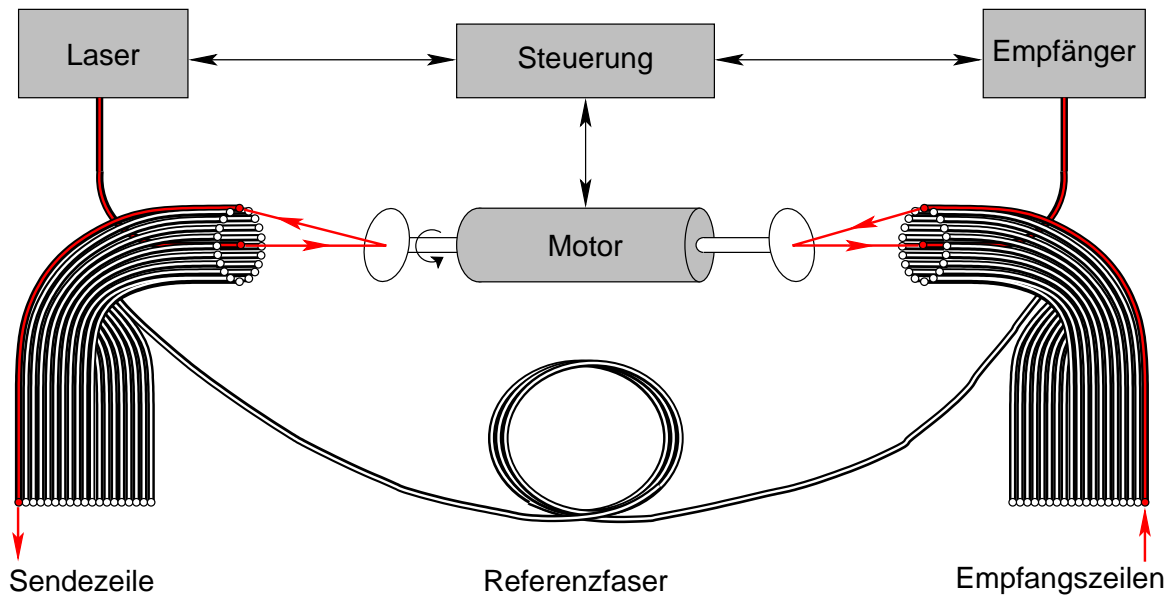


Abbildung 2.4: *Prinzip einer Scaneinheit der Firma TopoSys. Mit einem rotierenden, gekippten Spiegel wird das Laserlicht in ein Glasfaserbündel und von dort zum Boden gelenkt, wodurch das in Abbildung 2.3 gezeigte Abtastmuster entsteht.*

tationen vermessen, wobei der erste Puls an der Vegetationskrone, der letzte am Erdboden entsteht (s. Abbildung 2.5b).

Die Firma TopoSys ist inzwischen in der Lage, parallel zum Laserscanner mittels eines vierkanaligen Zeilensensors Farb- und Infrarotbilder zu erfassen. Da die Daten praktisch zeitgleich vom selben Standort erfasst werden, ist eine optimale Korrespondenz zwischen den vier Kanälen und dem Laserscan-Höhenmodell gegeben, so dass die aufwändige Referenzierung der multisensoriellen Daten untereinander entfallen kann.

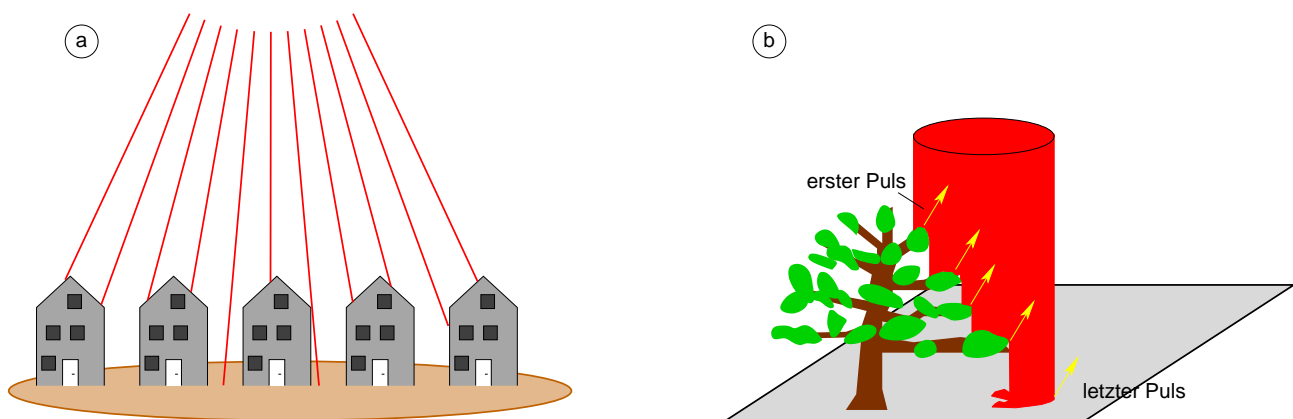


Abbildung 2.5: Wichtige Eigenschaften des Laserscanners: a) Je kleiner der Scanwinkel ist, desto steiler können die vermessbaren Kanten sein. Der TopoSys-Scanner mit einem Blickwinkel von $\pm 7^\circ$ erlaubt beispielsweise Steigungen von mehr als 800% zu vermessen. Dadurch ist es z.B. auch möglich, Häuserschluchten in dicht bebauten Innenstädten zu vermessen. b) Durch die Erfassung des ersten und letzten reflektierten Laserpulses, ist die gleichzeitige Vermessung von Baumkronen und Waldboden möglich.

2.2 Verfahren zur Segmentierung

Grundsätzlich kann man zwischen kanten- und regionenbasierten Segmentierungstechniken sowie Verfahren, die aufgrund eines physikalischen Modells segmentieren, unterscheiden. Die an einem physikalischen Modell orientierten Segmentierungsverfahren besitzen nur eingeschränkte Einsatzmöglichkeiten (s. [50][56]), wohingegen die kanten- und regionenbasierten Verfahren allgemein einsetzbar sind. Während die kantenbasierten Verfahren auf Grau- und Farbbildern angewendet in etwa die gleichen Ergebnisse liefern (s. [79][81]), lassen sich die regionenbasierten Verfahren durch die Verwendung von Farbbildern signifikant verbessern.

Die regionenbasierten Segmentierungsalgorithmen lassen sich bezüglich ihrer zugrundeliegenden Methodik in verschiedene Klassen einteilen. Haralick gibt in [48] einen sehr guten Überblick über die verschiedenen Methoden der regionenbasierten Segmentierung von Grauwertbildern. Dieselben Verfahren finden sich auch in den Veröffentlichungen zur Farbsegmentierung wieder (s. [98]) und lassen sich im Wesentlichen unterscheiden in Cluster-Analyse im Merkmalsraum, Region Growing, Split and Merge und Kombination aus diesen Verfahren (s. [88]), wie z.B. das auf dem Color Structure Code basierende Verfahren.

2.2.1 Cluster-Analyse im Merkmalsraum

Bei der Cluster²-Analyse im Merkmalsraum werden Pixel auf Merkmalsvektoren in einem Merkmalsraum abgebildet. Danach wird untersucht, ob die Merkmalsvektoren im Merkmalsraum Cluster bilden. Geht man davon aus, dass homogene Bildbereiche ähnliche Merkmalsvektoren besitzen und daher an diesen Stellen Häufungen im Merkmalsraum entstehen [31], kann man jedem Pixel die Marke (Label) des Clusters zuweisen, zu dem dieser gehört. Alle Pixel mit demselben Clusterlabel bilden dann ein Bildsegment. In der klassischen Mustererkennung gibt es sehr viele Methoden zur Detektion von Clustern mit jeweils eigenen charakteristischen Eigenschaften [54]. Häufig wird in der Bildsegmentierung die Histogrammanalyse zur Clusterdetektion eingesetzt. Bei einem eindimensiona-

²engl. Cluster = Anhäufung

len Merkmalsraum (z.B. bei eindimensionalen Bildern bei der Abbildung von Pixeln auf ihren Grauwert) erhält man die Cluster durch die Bestimmung der Intervalle von Werten zwischen den Tälern im Histogramm. Sollen Muster erkannt werden, ist es jedoch häufig notwendig die Beziehung zwischen mehreren Pixeln in einen mehrdimensionalen Merkmalsraum abzubilden. Hier kann es unter Umständen hilfreich sein, die gewünschten Cluster durch Analyse von Mustern vorab zu bestimmen.

Eine Weiterentwicklung dieser globalen Clusteranalyse ist die rekursive Anwendung des Verfahrens auf die segmentierten Teilbilder. In [82] wird das Verfahren für Farbbilder erweitert vorgestellt.

2.2.2 Region Growing

Beim Region Growing wird ausgehend von einer initialen Region das Verhältnis zu einem Nachbarpixel bestimmt und wenn dieses bezüglich bestimmter Eigenschaften der Region ähnlich ist, wird es in die Region aufgenommen. Die einfachste Variante des Region Growings ist die einfache Verkettung. Dabei wird jedes Pixel als ein Knoten eines Grafs betrachtet. Wird bei zwei benachbarten Pixeln die Schwelle eines Ähnlichkeitsmaßes überschritten, so werden die beiden Punkte mit einer Kante verbunden. Da eine Kante im Graf ausreicht, um zwei Regionen zu verschmelzen, ist diese Variante des Region-Growings für komplizierte Szenen nicht geeignet.

Bei der hybriden Verkettung werden die Nachbarschaftspixel in die Entscheidung mit einbezogen, ob zwei Pixel mit einer Kante verbunden werden und damit zu einer Region gehören. Dabei wird jedem Pixel aus einer $k \times k$ -Nachbarschaft ein Eigenschaftsvektor zugewiesen. Sind die Nachbarschaften ähnlich, werden die Pixel mit einer Kante verbunden.

Bei der zentroiden Verkettung werden nicht Paare von Pixeln miteinander verglichen, sondern der Wert eines Pixels wird mit dem Mittelwert eines nicht notwendigerweise fertigen Segments verglichen. Jedes hinzugefügte Pixel führt zu einer Veränderung des Segmentmittelwertes. Der Vorteil dieses Verfahrens ist, dass alle Pixel eines Segments in die Entscheidung mit einbezogen wer-

den, ob ein neues Pixel der Region hinzugefügt wird. Nachteilig ist jedoch, dass das Ergebnis vom Startpunkt der Segmentierung abhängig ist. Ein detaillierter Überblick über die Vor- und Nachteile der einzelnen Region-Growing-Verfahren wird in [106] gegeben.

2.2.3 Split and Merge

Im Gegensatz zu den Region-Growing-Verfahren, welche die Segmente aus kleinen Zellen zu größeren entwickeln, wird beim Split-and-Merge-Verfahren [52] zunächst der umgekehrte Weg beschritten. Zu Beginn wird das Bild in 2^n gleichgroße, quadratische Felder unterteilt. In der Split-Phase werden alle Felder, die ein bestimmtes Homogenitätskriterium nicht erfüllen, in 4 quadratische Felder aufgeteilt. Diese Teilung wird solange fortgesetzt, bis alle Felder das Homogenitätskriterium erfüllen. In der Merge-Phase werden benachbarte Felder verschmolzen, wenn sie gemeinsam das Homogenitätskriterium erfüllen. Auf diese Weise sind auch Regionen möglich, die sich über Quadrantengrenzen hinaus erstrecken.

2.2.4 Color Structure Code

Ein anderer Weg wird in [87][89] mit dem Color Structure Code, einem System zur schnellen Segmentierung von Farbbildern, beschritten. Das System arbeitet in einer hexagonalen, hierarchischen Inselstruktur (s. Abbildung 2.6), die in [49] beschrieben wird. Dazu wird jedes Pixel zusammen mit 6 seiner 8 Nachbarn in diese hexagonale Struktur überführt. In den Phasen Vorverarbeitung, Codierung und Verknüpfung wird eine Menge hierarchischer Datenstrukturen (Bäume) generiert, von denen jede ein zusammenhängendes farbiges Objekt darstellt.

In der Vorverarbeitung wird das Rauschen in den Bildern mittels eines Symmetric-Nearest-Neighbour-Verfahrens [84] unterdrückt. Das Besondere dieses Verfahrens ist, dass im Gegensatz zu einfachen Glättungsverfahren die Kanten in den Bildern erhalten bleiben, da eine Mittelwertbildung jeweils nur mit

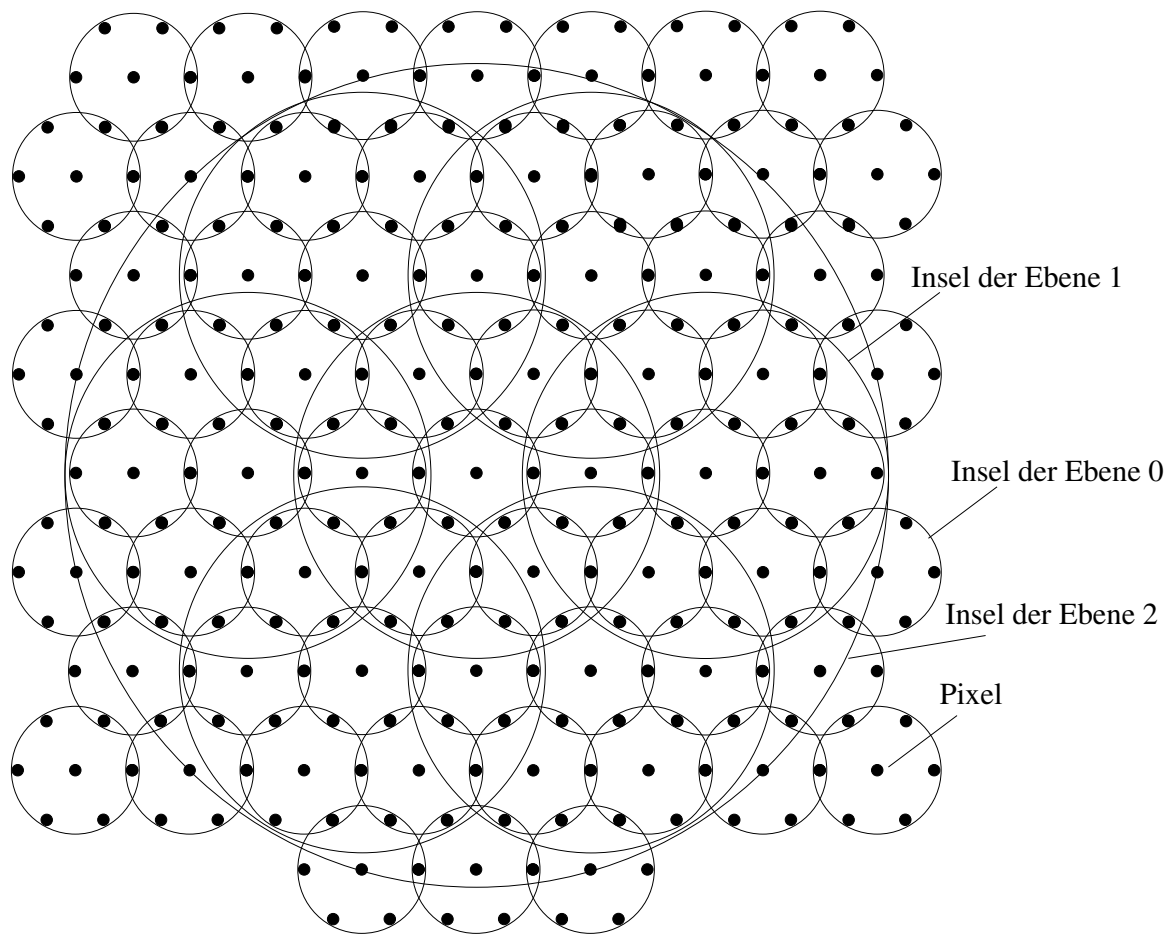


Abbildung 2.6: Aufteilung eines Bildes in eine hierarchische Struktur mit sich überlappenden hexagonalen Inseln.

der Hälfte der Nachbarpixel erfolgt, die dem Pixel bezüglich ihres Wertes am nächsten stehen.

In der Codierungsphase werden in der Ebene 0 jeweils die Bildpunkte einer Inselstruktur, die benachbart sind und bezüglich ihrer Farbe ein Ähnlichkeitsmaß erfüllen, zu einem Code-Element zusammengefaßt. Da immer mindestens zwei Pixel eine Region bilden müssen, können maximal 3 Code-Elemente entstehen.

In der Verknüpfungsphase werden die Code-Elemente der Ebene n zu Code-Elementen der Ebene $n + 1$ zusammengefaßt. Ein Element der Ebene n wird dann in ein Code-Element der Ebene $n + 1$ aufgenommen, wenn es zu einem benachbarten Code-Element der Ebene n farbähnlich ist und beide Elemente ei-

ne zusammenhängende Region bilden, was sich durch die Überlappungsbereiche der Inseln leicht verifizieren lässt. Angewendet auf alle Ebenen entsteht so für jede zusammenhängende Farbregion eine Baumstruktur. Das für lokale Regionwachstumsverfahren typische Problem von Verkettungsfehlern wird gelöst, indem auf jeder Verknüpfungsebene ein zusätzlicher Farbttest, der Farbmittelwerte zweier zusammenhängender Farbflächen vergleicht, durchgeführt wird. Liegt der Farbabstand über einer bestimmten Schwelle, werden die Farbregionen nicht verknüpft.

2.3 Wissensrepräsentationsformen

Eine Interpretation von Fernerkundungsdaten in dem Sinne, dass Bildinhalten eine Bedeutung zugewiesen werden soll, ist nur durch Einbringen von Wissen in den Interpretationsprozess möglich. Die Repräsentation des Wissens orientiert sich dabei an zwei unter Umständen gegensätzlichen Forderungen. Zum einen muss es möglich sein, das Wissen automatisch zu verarbeiten. Zum anderen entstammt das Wissen einem menschlichen Experten und muss von diesem formuliert und gewartet werden können.

Grundsätzlich wird unterschieden zwischen implizitem und explizitem Wissen[40][69]. *Implizites Wissen* steht direkt im Code eines Programms und beschreibt einen festen Ablaufplan. Der Algorithmus wird nicht abgearbeitet, weil eine bestimmte Methode gesucht wird, sondern weil zum Zeitpunkt des Erstellens feststand, welche Funktion in der vorliegenden Situation einzusetzen ist.

Sind dem System seine Problemlösungsmöglichkeiten für bestimmte Aufgaben bekannt, aus denen es für das vorliegende Problem eine passende Funktion auswählt, und sind diese im System an einer bestimmten Stelle explizit vermerkt, so spricht man von *explizitem Wissen*.

Bei explizitem Wissen ist wiederum zu unterscheiden zwischen deklarativem und prozeduralem Wissen. *Deklaratives Wissen* beschreibt, in welchem Verhältnis Objekte in der realen Welt zueinander stehen. Vorteil dieser Repräsentationsform ist ihre gute Lesbarkeit. Der Aufwand eine konkrete Lösung

zu bestimmen ist jedoch wesentlich höher als bei der Repräsentation durch *prozedurales Wissen*. *Prozedurales Wissen* beschreibt den Gebrauch von Wissen und gibt damit eine direkte Anleitung zum Lösen eines Problems. Die Abarbeitung von prozeduralem Wissen ist effizienter, dafür ist die Beschreibung des Wissens für komplexe Aufgabenstellungen sehr aufwändig und wenig wartungsfreundlich.

2.3.1 Wissensbasierte Systeme

Als wissensbasierte Systeme bezeichnet man Systeme, die ihr anwendungsspezifisches Wissen explizit und getrennt von einer allgemeinen Problemlösungsstrategie speichern. Für unterschiedliche Aufgabenstellungen wird lediglich die Wissensbasis ausgetauscht. Für die Repräsentation des Wissens in der Wissensbasis gibt es unterschiedliche Ansätze, von denen im Bereich der Interpretation von Fernerkundungsdaten hauptsächlich Regeln und semantische Netze verwendet werden.

Regeln Regelbasierte Systeme formulieren Wissensinhalte in Wenn-Dann-Beziehungen. Der Aktionsteil wird ausgeführt, wenn der Bedingungsteil erfüllt ist. Treffen mehrere Bedingungsteile zu, muss eine übergeordnete Instanz entscheiden, welche Regel anzuwenden ist. Umfangreiche Regelbasen werden schnell unübersichtlich. Logische Widersprüche innerhalb der Regeln sind dann nicht mehr auszuschließen.

Semantische Netze Ein semantisches Netz ist ein gerichteter Graf, in dem die Knoten Objekte, Ideen, Ereignisse oder Sachverhalte darstellen, die durch Kanten, welche die Beziehung der Knoten zueinander beschreiben, verbunden sind. Semantische Netze sind keine eigenständige Wissensrepräsentationsform. Sie dienen vielmehr der übersichtlichen Darstellung anderer Repräsentationsformen, die dann das Aussehen der Knoten und Kanten definieren. Für die Repräsentation von Wissen in der Fernerkundung werden beispielsweise häufig Frames in Form von semantischen Netzen dargestellt. Frames sind eine objektorientierte Form der Wissensrepräsentation. Es wird unterschieden

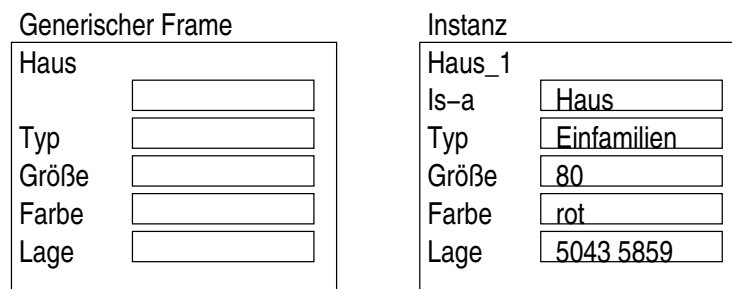


Abbildung 2.7: *Beispiel eines generischen Frames vom Typ Haus und einer Instanz. In der Instanz werden die Slots mit konkreten Werten belegt. Im Slot Is-a wird der Typ der Instanz gespeichert.*

zwischen generischen Frames, die ein Konzept eines Objekts der realen Welt darstellen, und Frame-Instanzen die für ein konkret vorhandenes Objekt der realen Welt stehen (s. Abbildung 2.7). Ein generischer Frame besitzt für jedes Attribut, mit dem ein Objekt beschrieben wird, einen Slot. Wird der generische Frame instanziiert, so werden den Slots der jeweilige Attributwert des konkreten Objekts zugewiesen. Die Beziehungen zwischen Objekten werden ebenfalls durch Slots hergestellt. Diese Beziehungsslots werden im semantischen Netz durch Kanten symbolisiert.

3 Konzept eines effizienten wissensbasierten Szenenanalysesystems

Allgemein betrachtet, gibt es für das Lösen komplexer Probleme zwei verschiedene Strategien:

1. Man beginnt mit der Lösung von Teilproblemen, in der Hoffnung diese irgendwann zu einer größeren Gesamtlösungsstrategie zusammenfassen zu können. Die Arbeit erfolgt somit vom Speziellen zum Generellen.
2. Die umgekehrte Strategie verfolgt einen strukturellen Ansatz. Es wird versucht, die groben Strukturen eines Problems zu erkennen und diese immer weiter zu verfeinern, in der Hoffnung, dass die Teilprobleme irgendwann so klein sind, dass sie lösbar werden (oder verschwinden). Hierbei handelt es sich um einen generalistischen Ansatz.

Auf die Interpretation von Fernerkundungsdaten angewendet, fallen in die erstgenannte Strategie alle Verfahren, die speziell auf die Detektion bestimmter Objektklassen ausgerichtet sind. Häufig verwenden diese Ansätze ein parametrisches Modell der gesuchten Objekte. Ist ein potentieller Kandidat im Bildmaterial gefunden, wird versucht die Modellparameter plausibel an die Daten anzupassen. Als Beispiel seien hier Verfahren zur Detektion von Gebäuden [20][97][104][103] und Straßen [5][12][14][24] genannt.

Der generalistische Ansatz ist in den wissensbasierten Systemen zur Interpretation multisensorieller Fernerkundungsdaten realisiert. Die Wissensbasis kann

dabei rein deklarativ als Faktenwissen verstanden werden sowie als prozedurales Wissen über die Nutzung dieser Fakten. Zahlreiche Formalismen zur expliziten Wissensrepräsentation sind in der Literatur vorgeschlagen worden [105]. Für die Interpretation von Fernerkundungsdaten werden Regeln und semantische Netze eingesetzt.

Als Beispiel für Systeme die *Regeln* für die *Analysesteuerung* verwenden seien hier ACRONYM [21], AIDA [67][43], BPI [70][100], ERNEST [80][93], MESSIE [30], MOSES [85][86], SIGMA [72], und SPAM [73] genannt. Die Systeme SPAM und BPI legen das *Wissen über die Szene* ebenfalls in *Regeln* ab, während die Systeme AIDA, ERNEST, MOSES, SIGMA und VISIONS [47] hierfür *semantische Netze* verwenden.

3.1 Eigenschaften bestehender Szenenanalyse-systeme

Da umfangreiche Regelbasen, wie sie für die Auswertung von Fernerkundungsdaten erforderlich sind, schnell unübersichtlich werden, sollen im Folgenden nur Systeme beispielhaft betrachtet werden, die ihr Wissen in Form von semantischen Netzen ablegen.

3.1.1 VISIONS

VISIONS steht für Visual Intergration by Semantic Interpretation of Natural Scenes. Das System wird seit 1975 an der Universität von Massachusetts in Amherst entwickelt. Es beinhaltet einen Ansatz für ein allgemeines Bildverarbeitungssystem, das in der Lage ist, komplexe Bilder wie beispielsweise Luftbilder automatisch zu verarbeiten. Die Vorverarbeitung und Segmentierung ist unabhängig von einer bestimmten Aufgabenstellung und arbeitet mit festen Algorithmen, einem regionenorientierten Verfahren nach [78][17] und einem kantenorientierten Verfahren nach [22][23]. Für die Interpretation der Bilder werden wissensbasierte Techniken verwendet. Die Bearbeitung statischer Luft-

bilder ist in [46][34] beschrieben.

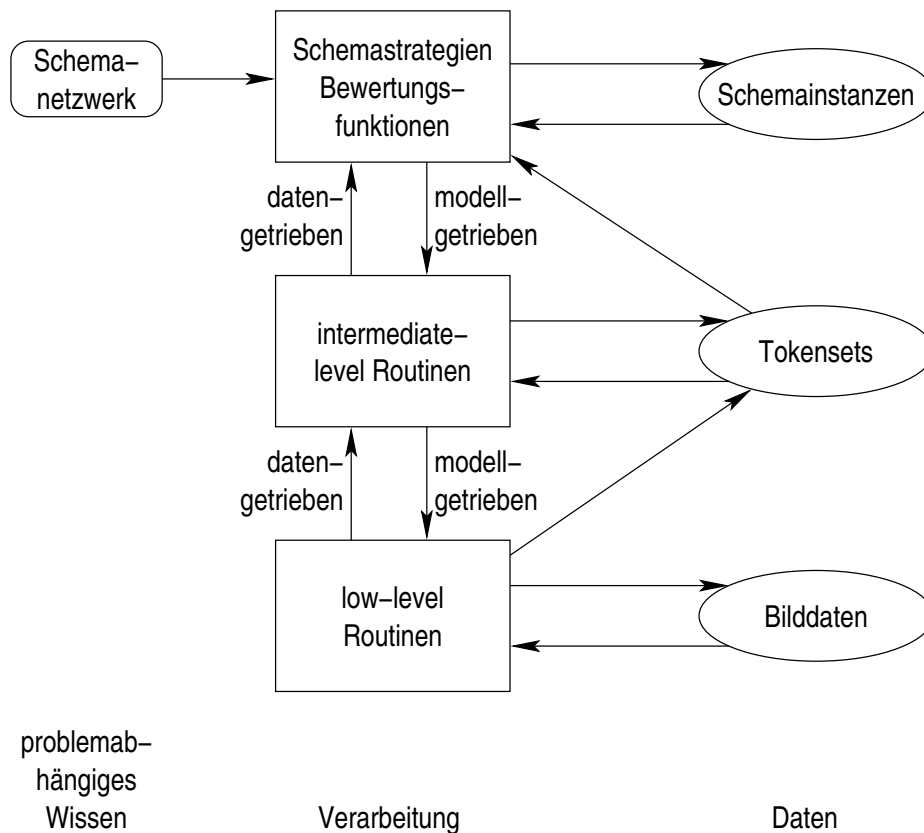


Abbildung 3.1: Die Architektur des VISIONS Systems gliedert sich in drei unterschiedlich abstrakte Ebenen. Die Bildverarbeitung findet auf unterster Ebene statt. Die mittlere Ebene fasst modell- und datengetriebene Primitive zu größeren Einheiten zusammen. Die oberste Ebene enthält das anwendungsspezifische Wissen und verwaltet die Instanzen.

Abbildung 3.1 zeigt die Architektur des VISIONS Systems. Sie gliedert sich in drei Abstraktionsebenen.

Auf der untersten Ebene wird datengetrieben gearbeitet. Das Bildmaterial wird segmentiert und einfache symbolische Beschreibungen sogenannter Tokensets generiert. Zu bestimmten Tokentypen werden jeweils bestimmte Merkmale archiviert.

In der mittleren Ebene werden die primitiven Tokensets zu komplexeren Tokensets zusammengefasst. Für diese Aufgabe stehen verschiedene Gruppen von

Verfahren bereit: merkmalsbasierte Klassifikation [91][66], perzeptives Gruppieren [90], Tokenbeziehungen [16][92], und modellgetriebene Resegmentierung [57]. Die Gruppierung zu größeren Tokensets geschieht sowohl datengetrieben nach Beendigung der untersten Ebene als auch modellgetrieben durch problemabhängige Kontrollstrategien.

Die oberste Ebene enthält das anwendungsspezifische Wissen in Form eines semantischen Netzes. Die Knoten des Netzes werden Schema genannt und repräsentieren im Bild vorkommende Objekte. Ein Schema enthält Restriktionen über zulässige Werte bestimmter Tokentypen. Wird ein Token dieses Typs mit passenden Werten gefunden, so wird eine Plausibilitätsprüfung durchgeführt, bei deren positiven Ausgang eine Verbindung zwischen dem gefundenen Objekt und dem Token hergestellt wird. Die so gefundenen Objekthypothesen werden als Schemainstanzen abgelegt.

3.1.2 ERNEST

ERNEST [60][61] wurde an den Universitäten Erlangen und Bielefeld entwickelt. Es handelt sich dabei um eine Sprache zur Wissensrepräsentation mit einer problemorientierten Kontrollstrategie basierend auf semantischen Netzen. Im Gegensatz zu allgemeinen semantischen Netzen verwendet ERNEST eine feste Anzahl von Knoten- und Kantentypen mit festgelegter Bedeutung. Das Knotenkonzept repräsentiert eine Klasse von Objekten. Eine Instanz ist direkt einem Teil des Signals zugeordnet. Das modifizierte Konzept stellt ein Zwischenergebnis der Analyse dar. Die Knoten sind mit einem von drei Kantentypen verbunden. Die Spezialisierung (*is-a*) entspricht dem Vererbungsmechanismus. Die Teil-von-Kante (*part-of*) zerlegt ein Konzept in seine Bestandteile. Die optionale Teil-von-Kante (*cdpart-of*) verbindet einen Konzeptteil, der nur in einem bestimmten Zusammenhang definiert ist. Um das Wissen in verschiedenen Abstraktionsebenen klar voneinander zu trennen, ist ein Kantentyp Konkretisierung-von (*con-of*) eingeführt worden.

Das Ziel der Analyse ist es, das Eingangssignal den Konzepten zuzuordnen, d.h. es sollen Instanzen der Konzepte generiert werden. Da das Signal oft mehrdeutig ist, sind konkurrierende Instanzen und modifizierte Konzepte zu erwarten, die

für verschiedene Interpretationsmöglichkeiten des Eingangssignals stehen. ERNEST benutzt den A*-Algorithmus, um eine Entscheidung bezüglich der besten Interpretation fällen zu können. Jeder Knoten des Suchraums repräsentiert eine in sich konsistente Interpretation des Sensorsignals und enthält damit die komplette Wissensbasis adaptiert auf das Signal. Die Inferenzen im Suchraum und damit die Generierung von Instanzen und modifizierten Konzepten basieren auf sechs problemunabhängigen Regeln. Beispielsweise kann eine Instanz nur generiert werden, wenn alle ihre Teile instanziiert sind. Die Regeln sind implizit im ebenfalls problemunabhängigen Steueralgorithmus integriert.

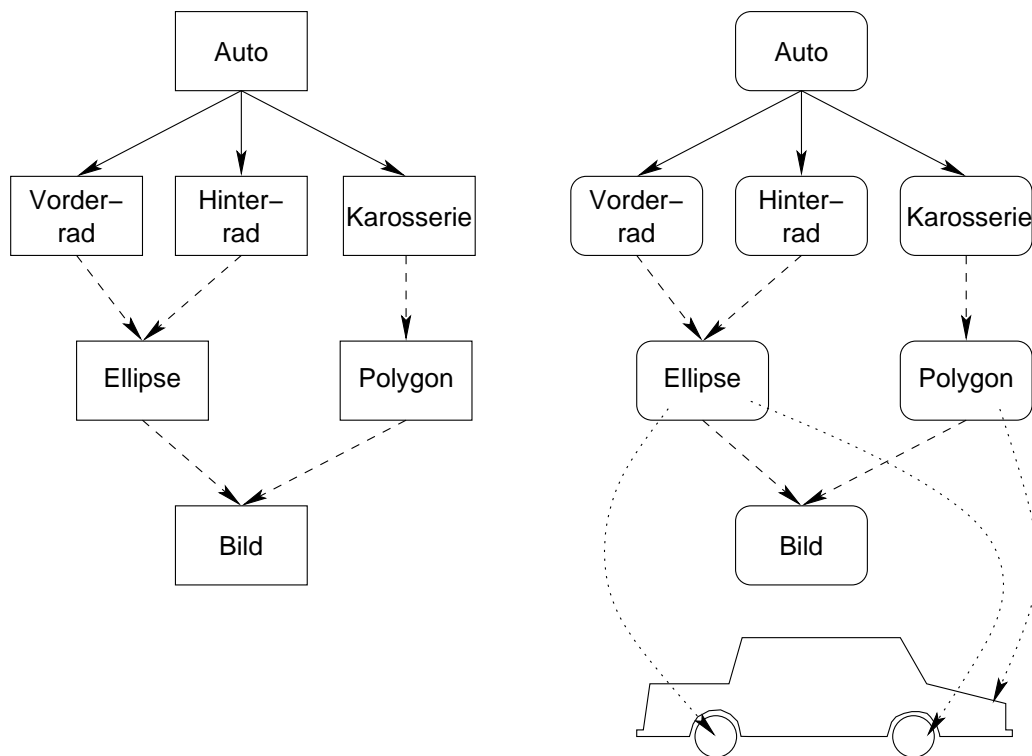


Abbildung 3.2: ERNEST: Einfaches Beispiel einer Wissensbasis (links) und instanzierter Wissensbasis (rechts) für ein Auto.

Abbildung 3.2 (links) zeigt beispielhaft eine einfache und sehr kleine Wissensbasis für ein System zur Erkennung eines Autos. In diesem Modell setzt sich das Auto aus den Komponenten Vorderrad, Hinterrad und Karosserie zusammen. Diese Teile werden den Bildprimitiven Ellipse und Polygon zugeordnet, die in einem Eingangsbild detektiert werden können. Abbildung 3.2 (rechts) zeigt die instanziierte Wissensbasis für eine konkrete Zuordnung von Instanzen zu einem

Eingangsbild.

3.1.3 AIDA

Das System AIDA (Automatic Image Data Analyzer) [67][101] wurde an der Universität Hannover zunächst für die wissensbasierte Auswertung von Luftbildern und frei gewählten Stereofotos [41] entwickelt und später für die Erkennung multitemporaler Strukturen [42] erweitert. In Abbildung 3.3 ist die Architektur von AIDA dargestellt. Als Eingangsdaten akzeptiert AIDA verschiedene Sensordaten wie RMK und SAR. Das Wissen über die zu interpretierende Szene wird in einem generischen Szenenmodell verwaltet. Informationen über die zu untersuchende Szene können auch einem GIS¹ entnommen werden, wobei speziell eine Anbindung an das in Deutschland verwendete ATKIS² [62][44] implementiert wurde. Im Bildverarbeitungsmodul erfolgt die Segmentierung der Sensordaten mit anschließender Extraktion von Bildprimitiven. Diese Bildprimitive der unterschiedlichen Sensoren werden in der symbolischen Verarbeitungseinheit fusioniert und zu komplexeren Strukturen gruppiert. Diesen wird eine symbolische Bedeutung zugewiesen. Neben dieser datengetriebenen Strategie kann die Analyse auch modellgetrieben erfolgen. Dabei leitet das System aus dem generischen Szenenmodell Hypothesen über die in den Sensordaten zu erwartenden Objekte ab und verifiziert diese anschließend. Das System liefert schließlich eine symbolische Szenenbeschreibung.

Ebenso wie ERNEST benutzt auch das System AIDA ein semantisches Netz als Wissensbasis (s. Abbildung 3.4). Dieses besteht aus Knoten und Kanten. Die Knoten speichern das Modellwissen über in der Szene zu erwartende Objekte. Die Kanten modellieren die Art der Beziehung zwischen den Knoten. In AIDA wird zwischen *hierarchischen* und *topologischen* Kanten unterschieden.

Hierarchische Kanten beschreiben die Beziehung zwischen zwei Knoten in übereinanderliegenden Ebenen der Hierarchie. AIDA unterstützt die folgenden Typen:

¹Geoinformationssystem

²Amtlich Topographisch-Kartographisches Informationssystem

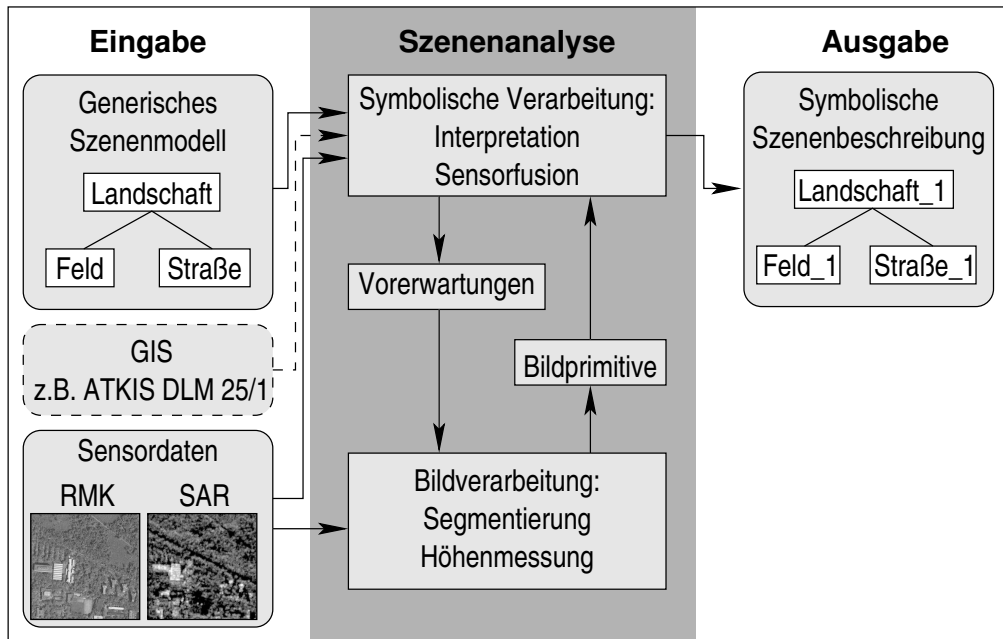


Abbildung 3.3: Architektur des wissensbasierten Bildinterpretationssystems AIDA: Die Eingangsdaten werden in der Szenenanalyse zu einer symbolischen Szenenbeschreibung aufbereitet.

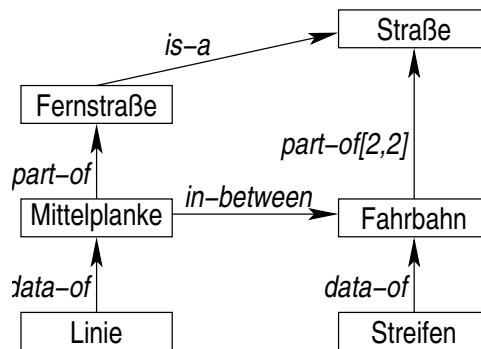


Abbildung 3.4: AIDA unterscheidet im semantischen Netz zwischen verschiedenen Kantentypen. Hierarchische Kanten beschreiben Beziehungen zwischen Objekten unterschiedlicher Ebenen in der Hierarchie (z.B. Fernstraße ist eine Straße). Topologische Kanten beschreiben Beziehungen zwischen Objekten gleicher Hierarchieebene (z.B. Mittelplanke befindet sich zwischen Fahrbahnen).

is-a: Die klassische Vererbung wird mit der *is-a*-Kante erzielt. Der Kindknoten erbt alle Eigenschaften seines Elternknoten.

part-of: Der Kindknoten ist Teil eines Elternknoten. Eine minimale und maximale Anzahl von Kindknoten dieses Typs kann angegeben werden. Die Differenz zwischen Maximum und Minimum beschreibt die Anzahl der optionalen Knoten.

cdpart-of: Manche Teilobjekte erfordern das Vorhandensein einer bestimmten Kombination von anderen Teilobjekten, d.h. sie benötigen einen speziellen Kontext. Diese Kanten werden über eine *context-dependend-part-of*- oder kurz *cdpart-of*-Kante angebunden.

concrete-of: Symbolische Bedeutungen müssen konkreten Geometrien im Bild zugeordnet werden. Den Übergang zwischen zwei konzeptionellen Ebenen beschreibt die *concrete-of*-Kante.

data-of: Mittels der *data-of*-Beziehung werden die Datenknoten der Bildverarbeitungsebene angebunden.

Als Erweiterung gegenüber anderen auf semantischen Netzen basierenden Interpretationssystemen führt AIDA *topologische Kanten* ein. Topologische Kanten beschreiben die Beziehung zwischen Objekten derselben Hierarchieebene. Die Art der topologischen Relation wird dabei über die der Kante zugeordneten Attribute bestimmt. Die Kante wird daher als *attributierte Relation* oder kurz *attr-rel* bezeichnet.

Wie auch in VISIONS und ERNEST ist die Bildverarbeitung Teil des Systems und wird auf unterster Ebene ausgeführt. Neue Bildverarbeitungs-routinen können über die Scriptsprache Tcl eingebracht werden. Mit den in der Bildverarbeitungsebene generierten Bildprimitiven wird dann rein strukturell weitergearbeitet. Die Auswahl der besten Strukturkombination erfolgt global über einen A*-Algorithmus.

3.2 Konzept des neuen Szenenanalysesystems GeoAIDA

Die hier vorgestellten Systeme haben viele gemeinsame Eigenschaften. So verwenden alle Systeme ein semantisches Netz zur Repräsentation des Modellwissens. VISIONS verbindet die Knoten des Netzes über *part-of*-Kanten. Querverbindungen zwischen den Knoten sind über Schemastrategien möglich, die für jedes Problem neu erarbeitet werden müssen. ERNEST stellt zwei weitere Kantentypen *is-a* und *con-of* zur Verfügung, die einen Vererbungsmechanismus implementieren bzw. die Verbindung von Knoten in unterschiedlichen Abstraktionsebenen erlauben. AIDA erweitert das semantische Netz schließlich um die Darstellung von Nachbarschaftsbeziehungen innerhalb einer Hierarchieebene mittels einer Kante, die attributierte Relationen beschreibt. Die Anbindung von Bildprimitiven erfolgt über eine Kante *data-of*. Sowohl ERNEST als auch AIDA verwenden zur Suche der besten Lösung einen A*-Algorithmus, d.h. es wird jeweils die vielversprechendste Lösungsmenge bevorzugt bearbeitet. Beim System VISIONS ist die Bildverarbeitung fester Bestandteil. VISIONS und AIDA binden die Daten jeweils nur an den Blattknoten des semantischen Netzes an. In höheren Ebenen wird mit diesen Knoten strukturell weitergearbeitet.

Für die Auswertung von Fernerkundungsdaten erweist sich AIDA als das geeignetste System, da es insbesondere auch die Anbindung von Vorwissen aus einem GIS erlaubt. Nachteilig ist jedoch, dass die Datenanbindung auf unterster Ebene des semantischen Netzes erfolgt und danach rein strukturell gearbeitet wird. Bei den in der Fernerkundung üblichen großen Eingangsbildern entstehen sehr große Mengen an Bildprimitiven, die über den A*-Algorithmus zu der am besten zum Modell passenden Lösung gruppiert werden müssen. Obwohl der A*-Algorithmus jeweils die vielversprechendste Teillösung aus der Menge aller möglichen Kombinationen weiterverfolgt, hat sich das Verfahren in der Praxis für die Verarbeitung großer Bilder wegen zu langer Rechenzeiten nicht bewährt.

Für den Einsatz im Bereich der Geodatenanalyse wurde daher ein Nachfolgesystem entwickelt, das die aus der Arbeit mit AIDA gewonnenen Erkenntnisse umsetzt und für die Verarbeitung von Fernerkundungsdaten optimiert ist. In

Anlehnung an das Haupteinsatzgebiet des neuen Systems wird das Programmpaket unter dem Namen GEOAIDA geführt. Die Unterschiede zwischen den beiden Systemen sind im Folgenden kurz dargestellt:

- GEOAIDA selbst besteht im Kern nur noch aus der Verwaltung eines semantischen Netzes und der Ablaufsteuerung der Analyse. Während der Analyse startet GEOAIDA Operatoren aus einem Pool von externen Programmen, welche die eigentliche Segmentierung, Klassifizierung und Bewertung übernehmen. Die Fernerkundungsdaten liegen geokodiert in einem Bilddatenpool. Aus diesem Bilddatenpool generiert GEOAIDA die für die Ausführung eines Operators notwendigen Teilbilder.
- Durch die vollständige Ausgliederung der Bewertung in externe Operatoren, ist die Implementierung spezieller Relationen zwischen den Knoten des semantischen Netzes in GEOAIDA überflüssig. Waren in AIDA für die Einführung neuer Relationen wie z.B. „in der Nähe von“ Änderungen am System nötig, übernimmt in GEOAIDA diese Aufgabe ein externes Programm, das beliebige Relationen zwischen den ihm übermittelten Knoten prüfen kann. Für verschieden Relationsarten gibt es dabei für den jeweiligen Einsatzzweck optimierte Programme.
- In AIDA erfolgt die Datenanbindung konzeptionell auf unterster Ebene. Die Analyse fasst dann die im Bild erkannten Primitive zu größeren Strukturen zusammen. Die Auswahl alternativer, modellkonformer Strukturen erfolgt dabei global über einen A^* -Algorithmus. Dies führt bei einer hohen Zahl von Primitiven jedoch zu sehr langen Rechenzeiten. In GEOAIDA kann die Datenanbindung in jedem Knoten des semantischen Netzes erfolgen (s. Abbildung 3.5). So genannte holistische Operatoren segmentieren die Bilddaten in kleinere Regionen. Diese Regionen werden dann wiederum von regionentypspezifischen Operatoren weiter verarbeitet. Existiert in einem Knoten kein Segmentierungsoperator, so wird konventionell über die Bestandteile des Knotens (strukturell) weiter analysiert.
- In AIDA wird die eigentliche Bildverarbeitung in der Scriptsprache *Tcl* durchgeführt. GEOAIDA bindet die gesamte Bildverarbeitung und Bewertung über eigenständige Programme ein, die in einer beliebigen Pro-

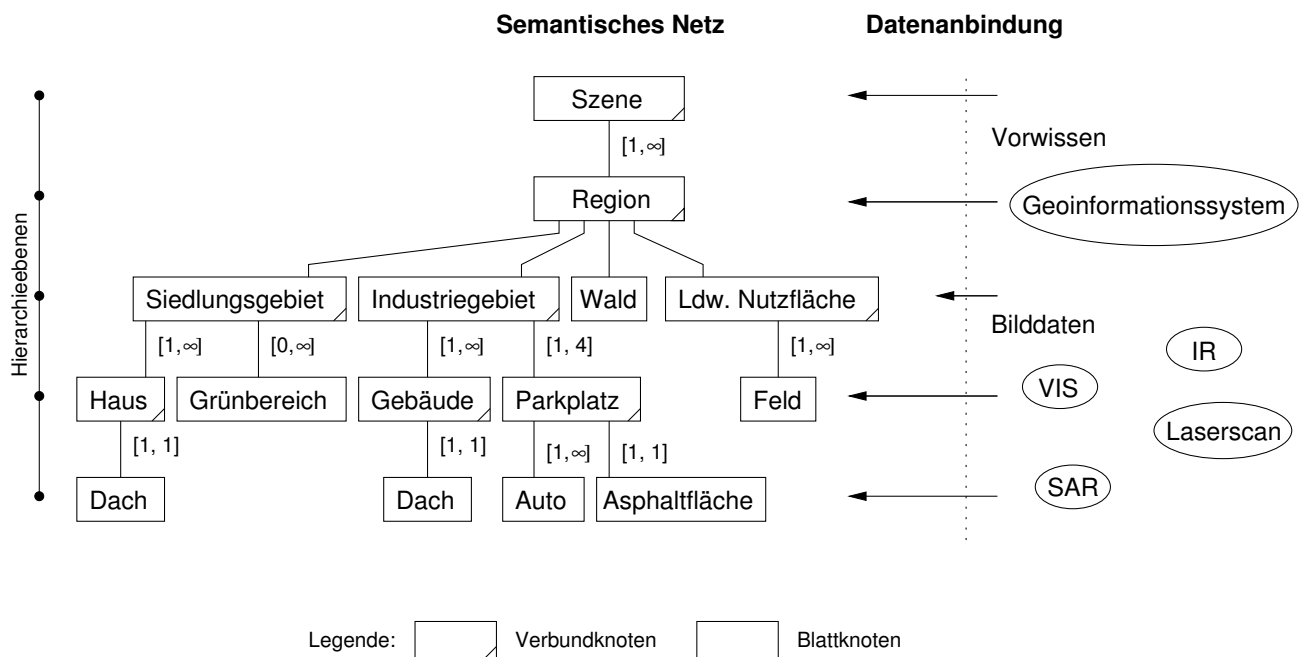


Abbildung 3.5: In GEOAIDA können Daten in jeder Hierarchieebene des semantischen Netzes angebunden werden.

grammiersprache geschrieben sein können. Dadurch ist es möglich, zum einen effiziente Programme in compiler-basierten Sprachen, zum anderen auch kommerzielle Programme zu nutzen, für die kein Quelltext vorliegt. Dies schließt unter anderem auch die zu Beginn dieses Kapitels erwähnten Speziallösungen für die Detektion bestimmter Objektklassen ein.

- GEOAIDA ist für die Interpretation von Fernerkundungsdaten ausgelegt. Jedem Objekt des Ergebnisnetzes ist deshalb eine Region im 2D-Raum zugeordnet.

Der Entwurf des semantischen Netzes geschieht in GEOAIDA unter zwei Gesichtspunkten:

- basierend auf der gewünschten Objekthierarchie, wie sie beispielsweise durch ein GIS vorgegeben ist, und
- basierend auf einer möglichen Lösungsstrategie.

Problemlösungsstrategie Die Auswertung von Fernerkundungsdaten ist ein sehr komplexes Problem. Das übliche Vorgehen zur Lösung komplexer Probleme ist die Zerlegung des Problems in möglichst voneinander unabhängige Teilprobleme. Dabei können die drei folgenden Fälle auftreten (s. Abbildung 3.6)

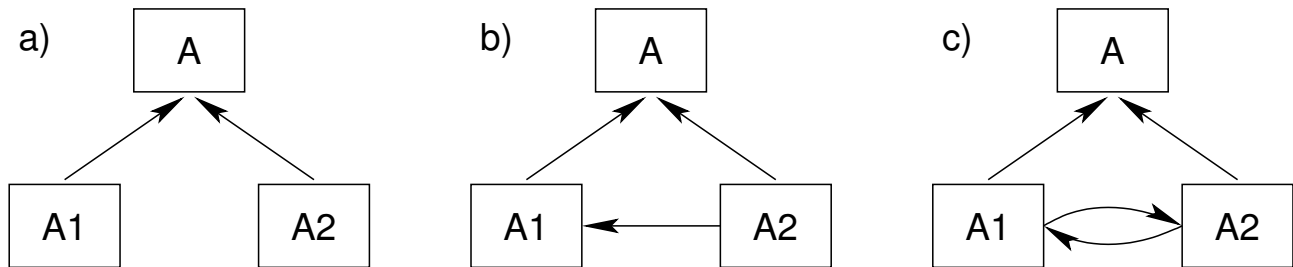


Abbildung 3.6: Zerlegung eines Problems in Teilprobleme: a) Die Probleme A1 und A2 sind voneinander unabhängig, können einzeln gelöst werden und führen zur Gesamtlösung des Problems A. b) A1 ist abhängig von A2. Für die Lösung des Problems A ist daher zunächst die Lösung des Problems A2 notwendig, woraufhin mit der Lösung des Problems A1 die Gesamtlösung A erbracht werden kann. c) A1 und A2 sind in beide Richtungen voneinander abhängig. Das Gesamtproblem A lässt sich dann in der Regel nur iterativ lösen.

1. Die Teilprobleme sind voneinander unabhängig lösbar (Abbildung 3.6a). Die Lösungsmenge des Gesamtproblems ergibt sich aus der Vereinigungsmenge der einzelnen Lösungsmengen, wobei bei nicht disjunkten Teillösungsmengen eine Entscheidung zugunsten einer Teillösung getroffen werden muss. Bezogen auf die Interpretation von Fernerkundungsdaten bedeutet dies, dass zwei Objektklassen voneinander unabhängig erkannt werden können. Die Verarbeitung kann parallel erfolgen.
2. Die Teilprobleme sind einseitig voneinander abhängig (Abbildung 3.6b), aber bilden keine Ringstruktur. Die Teilprobleme werden in der Reihenfolge der Abhängigkeiten gelöst. Dabei wird im Folgeproblem in der Regel nur noch der freie Lösungsraum untersucht. Lösungsmengenüberschneidungen müssen wie im ersten Punkt aufgelöst werden.

3. Die Teilprobleme sind gegenseitig voneinander abhängig oder bilden Ringstrukturen. Die Lösung kann in diesem Fall i.d.R. nur auf iterativem Wege gefunden werden. Konkret bedeutet dies, dass die Erkennung der beiden (oder aller) Objektklassen innerhalb einer Analyse erfolgt.

In allen drei Fällen können Überschneidungen der Lösungsmengen auftreten. Es ist daher wünschenswert, dass die Teilproblemlöser eine Gütebewertung ihrer Klassifizierung durchführen. aufgrund dessen bei der Fusion von Teillösungen eine Konfliktauflösung stattfinden kann. Sowohl die Teillösung als auch die Fusion von Teillösungen wird in GEOAIDA über externe Programme abgewickelt.

Ergebnisvisualisierung Bedingt durch die große Menge an Eingangsdaten sind auch die errechneten Ergebnisse äußerst komplex. Es stellt sich daher die Frage, wie diese Ergebnisse einem Anwender in geeigneter Form präsentiert werden können. GEOAIDA stellt hierfür eine aus den drei Komponenten semantisches Netz, Instanzennetz und Labelbilder, welche die Lage der Objekte beschreiben, bestehende interaktive Karte bereit. Das semantische Netz sowie das mit Hilfe des Datenmaterials generierte Instanzennetz werden in Form eines Baumes dargestellt, in dem sich einzelne Zweige ausblenden lassen. Parallel hierzu werden in einer Karte die aktuell in den Netzen sichtbaren Objekte eingezeichnet. Als Hilfestellung lassen sich die Originaldaten der Karte überlagern. Umgekehrt lassen sich in der Karte Objekte auswählen, deren Eigenschaften dann im Instanzennetz evaluiert werden können.

4 Architektur des wissensbasierten Szeneninterpretationssystems GeoAIDA

Das System GEOAIDA besteht aus mehreren Komponenten, die unabhängig voneinander genutzt werden können (s. Abbildung 4.1). Auf unterster Ebene werden C- und C++-Bibliotheken zur Verfügung gestellt, die das Einlesen und die Manipulation von Bildern sowie die symbolische Verarbeitung von Objektmengen erlauben. Häufige Verarbeitungsschritte sind in kleinen ausführbaren Programmen implementiert, die als parametrisierte Filter arbeiten. Aus der Kombination mehrerer Filter ergeben sich Operatoren, wie sie im wissensbasierten Bilddatenanalysator GEOAIDA benötigt werden. Zur Bewertung von Bilddaten werden einige GUI-basierte Werkzeuge bereit gestellt. Ein Experimentalsystem zum Entwurf von Operatoren sowie der Bilddatenanalysator GEOAIDA mit integrierter 2D-Visualisierungseinheit bilden die Highlevel-Werkzeuge.

4.1 Basis-Bibliotheken für die Verarbeitung von Fernerkundungsdaten

In den Basisbibliotheken werden verschiedene, immer wieder benötigte Funktionen bereitgestellt. Es wurden Bibliotheken für die Ablage von Fernerkundungsdaten, für die Manipulation von Bildern unterschiedlichen Datentyps, für die Umrechnung zwischen Bild- und Weltkoordinaten sowie für die Verarbeitung

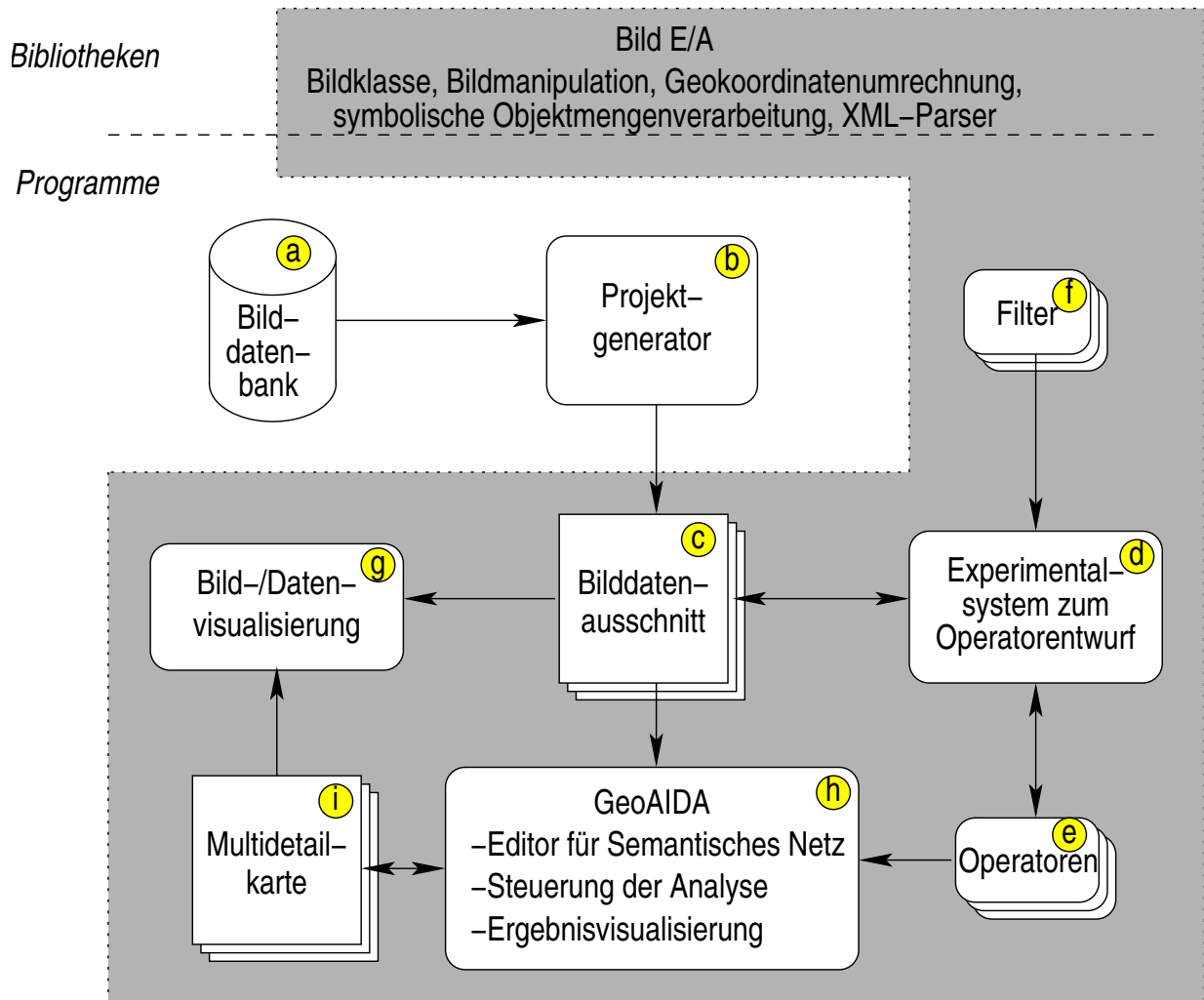


Abbildung 4.1: Projektübersicht zu GEOAIDA: Das Bildmaterial wird in einer Datenbank (a) gehalten. Über den Projektgenerator (b) können Bildausschnitte (c) ausgewählt werden. Das Experimentalsystem (d) ermöglicht das interaktive Erstellen von Operatoren (e) aus selbst erstellten oder vorhandenen Filterprogrammen (f) oder den schon bestehenden Operatoren (e). Mittels Bild-/Datenvisualisierungsprogrammen (g) können sowohl die Ergebnisse (c) des Experimentalsystems (d) bzw. des Projektgenerators (b) als auch die Ergebnisse (i) von GEOAIDA (h) ausgewertet werden. Für die Bildverarbeitung verwenden alle Programme gemeinsame Bibliotheken. Die grau hinterlegten Bereiche sind Teil dieser Arbeit.

von Objektmengen entwickelt. Für das Einlesen von in einer an die XML¹ angelehnten Sprache abgelegten Objekten und deren Attribute steht ein Parser zur Verfügung.

4.1.1 Ein einfaches Datenformat zum Ablegen von Fernerkundungsdaten

Fernerkundungsdaten liegen in der Regel als zweidimensionale Datenfelder vor. Um eine hinreichende Genauigkeit der Daten zu erreichen, überschreitet der Dynamikbereich jedoch oftmals einen 8bit-Wertebereich. Da die Daten häufig aus verschiedenen Datenquellen stammen, liegen sie nicht in einheitlichen Formaten vor. In der Regel lassen sie sich jedoch in ein einfaches Rasterdatenformat überführen, in dem die Daten zeilenweise abgelegt sind. Für die Lokalisierung der Daten ist es zudem wichtig, die Geokoordinaten sowie deren Auflösung zu vermerken.

Übliche Bilddatenformate sind Rasterformate, welche die Daten ebenfalls zeilenweise ablegen. Jedoch unterstützen die meisten Bildformate nur 8bit-Werte oder erfordern beim Überschreiten dieses Wertebereichs eine bestimmte Byteorder.

Verschiedene Bildverarbeitungsalgorithmen setzen oftmals bestimmte Wertebereiche und Datentypen voraus, so dass die Bilddaten häufig zwischen verschiedenen Formaten konvertiert werden müssen.

Für die Arbeit mit Fernerkundungsdaten wurde daher ein Datenformat entwickelt, das rechnerarchitekturunabhängig ist, die Wertebereiche 8bit, 16bit, 32bit und Floating-Point, Skalierung und Translation sowie eine Typwandlung beim Einlesen der Daten unterstützt.

Das Format selbst besteht aus einem im Klartext lesbaren Kopf (ASCII²-Header), worauf unmittelbar folgend die Rasterdaten zeilenweise binär oder im Klartext abgelegt sind. Dieses Format hat den Vorteil, dass der Header

¹eXtended Markup Language

²American Standard Code for Information Interchange

mit einem beliebigen Editor erstellt werden kann. Eine gültige Datei entsteht dann durch einfaches Aneinanderkopieren des Headers und der Rohdaten. Der Header enthält Informationen über den Datentyp, die Größe und den Minimal-/Maximalwert der Daten. Zusätzlich können Geokoordinaten sowie frei wählbare Kommentare eingefügt werden. Die gesamte Datei kann mit gängigen Komprimierungsprogrammen in ihrer Größe reduziert werden.

Für die Verarbeitung dieses Datenformats wird eine C-Bibliothek zur Verfügung gestellt, die das Lesen und Schreiben der Daten unterstützt. Beim Einlesen können die Daten dabei automatisch in verschiedene Formate überführt werden. Für spezielle Anwendungsfälle ist es sogar möglich eine Transferfunktion vorzugeben, die auf jedes Datum angewendet wird. Da das hier beschriebene Datenformat dem weit verbreiteten PAM-Format³ sehr ähnlich ist, wird das Lesen und Schreiben dieses Formats ebenfalls unterstützt.

4.1.2 Bildklasse für die Manipulation von Fernerkundungsdaten

Fernerkundungsdaten können in unterschiedlichen Datentypen vorliegen. So werden Grauwertbilder oftmals in 8bit großen Datenstrukturen abgelegt, wohingegen Höhenbilder üblicherweise in 16bit oder Float-Format abgelegt werden. Ein Experimentalsystem wie GEOAIDA sollte einerseits in der Lage sein, diese Datentypen zu verarbeiten, andererseits aber die Übertragung von Algorithmen, die für einen bestimmten Datentyp erstellt wurden, auf einen anderen Datentyp ermöglichen, ohne dass die entsprechenden Programme geändert oder compiliert werden müssen.

Um schnell von der Idee zum Prototypen zu gelangen, sollte die Anwendung von Bildtypen weitgehend der von einfachen Datentypen entsprechen. Die Verwaltung des für die Bilder notwendigen Speichers sollte in der Bildklasse gekapselt sein. Angesichts der in der Fernerkundung üblichen großen Datenmengen ist unnötiges Vervielfältigen von Daten zu vermeiden.

Im System GEOAIDA wird dies mit der in der Abbildung 4.2 gezeigten Klassen-

³Portable-Any-Map-Format

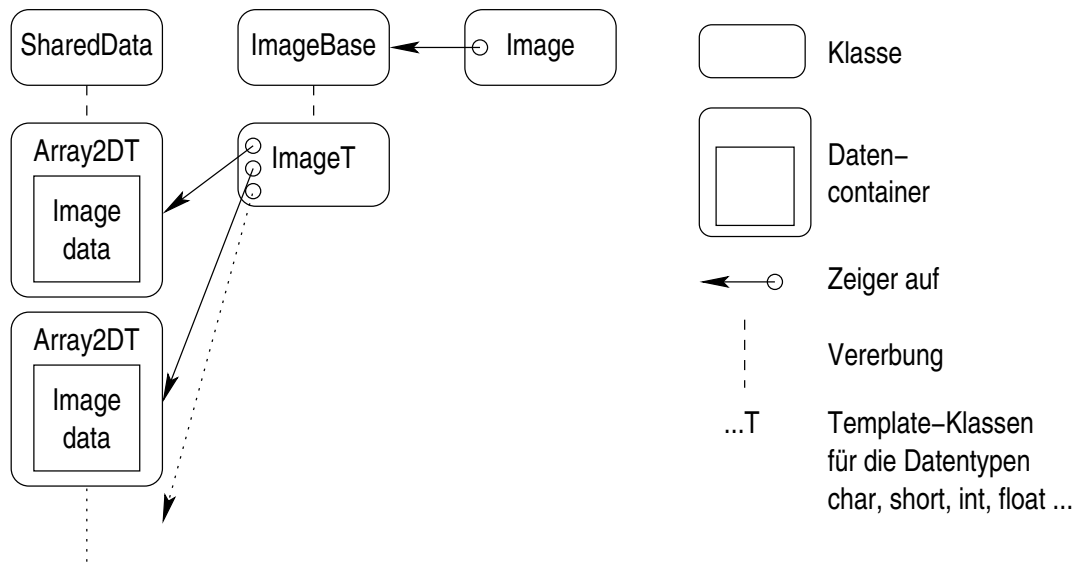


Abbildung 4.2: In der GEOAIDA-Bildklasse sind die Datentypen vollständig gekapselt. Für einen bestimmten Datentyp entwickelte Funktionen können auf diese Weise zweckentfremdet auf andere Datentypen angewendet werden. Durch die Verwendung von impliziten Daten-Sharing kann mit Bildern genauso einfach gearbeitet werden, wie mit einfachen Datentypen, ohne dass dabei mehr Speicher als nötig verbraucht wird.

struktur erreicht. Die Datentypunabhängigkeit ergibt sich durch eine Basisklasse `ImageBase`, die alle Methoden der typspezifischen Klassen als virtuelle Methoden enthält. Von dieser werden die typspezifischen Klassen `ImageT<type>` abgeleitet, wozu sich die Anwendung von Templates empfiehlt, da sich die Quelltexte der typspezifischen Klassen nur in den verwendeten Datentypen unterscheidet. Die typspezifischen Klassen `ImageT<type>` enthalten jeweils entsprechend der Anzahl der Kanäle Zeiger auf typabhängige zweidimensionale Datencontainer `Array2DT<type>`, welche die eigentlichen Bilddaten aufnehmen. Diese Datencontainer sind von der Klasse `SharedData` abgeleitet, wodurch die automatische Speicherverwaltung realisiert ist. Die Klasse `SharedData` verwaltet einen Referenzzähler. Das Kopieren der Daten z.B. bei einer Zuweisung erfolgt durch einfaches Kopieren des Zeigers auf die eigentlichen Daten und Inkrementieren des Referenzzählers. Soll nun wirklich schreibend auf die Daten zugegriffen werden, wird überprüft, ob mehr als eine Referenz auf die Daten besteht. Ist dies der Fall,

muss eine Kopie der Daten angelegt werden. Lesender Zugriff ist grundsätzlich von beliebig vielen Stellen auf den gleichen Datenbereich möglich.

Die beschriebene Klassenstruktur erlaubt nun eine datentypunabhängige Verarbeitung von Bildern. Der Datentyp muss jedoch beim Anlegen eines Bildes bekannt sein und kann danach nicht mehr verändert werden. Wird das Bild z.B. aus einer Datei gelesen, ist der Datentyp jedoch nicht in jedem Fall im Voraus bekannt. Andererseits lässt sich die typlose virtuelle Basisklasse *ImageBase* aber nicht anlegen. Den Ausweg bietet eine Rahmenklasse *Image*, die einen Zeiger auf die virtuelle Basisklasse *ImageBase* enthält. Die Rahmenklasse lässt sich zunächst typunabhängig anlegen. Erst wenn der Typ gesetzt wird, z.B. durch Einlesen von Bilddaten, wird die entsprechende typspezifische Klasse *ImageT<type>* angelegt. Der Klasse *Image* müssen natürlich alle Datentypen bekannt sein. Soll die Menge der verarbeitbaren Datentypen erweitert werden, so sind Ergänzungen in dieser Klasse notwendig. In der Beschränkung der Datentypen liegt jedoch andererseits der Vorteil, dass im Gegensatz zu einer reinen Template-Klassenbibliothek eine Bibliothek vorab compiliert werden kann.

Die Bildklasse enthält alle grundlegenden Bildmanipulationsfunktionen, vom Zugriff auf einzelne Pixel bis hin zu unterschiedlichen Algorithmen für das Resamplen von Bildern. Geht man von mehr als einem Nutzer der Bildklasse aus, sollte neue Funktionalität in der Regel nicht in Form von Methoden in der Ursprungsklasse *Image* oder in davon abgeleiteten Klassen integriert werden. Wenn Nutzer *A* seine Methoden in der abgeleiteten Klasse *ImageA* implementiert und Nutzer *B* entsprechend in der Klasse *ImageB*, so sind die entsprechenden Methoden für den jeweils anderen Nutzer nur in einer weiteren, sowohl von der Klasse *ImageA* als auch von der Klasse *ImageB* abgeleiteten Klasse nutzbar. Es sollte daher von einer schon in der STL⁴ [75] genutzten Methode Gebrauch gemacht werden, d.h. es werden Funktionen geschrieben, die auf die Bildklasse angewendet werden. Dadurch ist es möglich, Funktionen aus verschiedenen Quellen zu kombinieren. Während in der STL Funktionen als reine typkonfigurierbare Template-Funktionen realisiert werden, gibt es mit der Bildklasse zwei Möglichkeiten der Implementierung. Die eine Möglichkeit besteht in einer Implementierung mittels der bereitgestellten typunabhängigen

⁴Standard Template Library

Methoden der *Image*-Klasse. Eine ausführungstechnisch effizientere Methode ist die Implementierung als Template-Funktion auf die Klasse *ImageT<type>*. Dies entspricht dem Vorgehen in der STL. Zusätzlich sollte in einer typunabhängigen Funktion über einen bereitgestellten Makro die Template-Funktion für alle Datentypen instanziiert werden. Diese neue typunabhängige Funktion kann nun von allen Nutzern verwendet werden, ohne dass sie Änderungen an ihren eigenen Funktionen vornehmen müssen. Diese zweite Methode eignet sich besonders für lauffzeitkritische Anwendungen, z.B. Operationen, die pixelweise ausgeführt werden. Auf diese Weise wurden z.B. Gradienten- und Varianz-Funktionen implementiert.

4.1.3 Geo- und Bildkoordinaten

Das Bildanalysesystem GEOAIDA arbeitet intern mit einem geodätischen Bezugssystem. Dies bietet den Vorteil, dass eine genaue Lagebeschreibung des verwendeten Bildmaterials sowohl in Beziehung zur Außenwelt als auch von Bildausschnitten untereinander möglich ist. Die Umrechnung von Bild- in Geokoordinaten und umgekehrt ist dabei verhältnismäßig einfach. Zu beachten ist allerdings, dass ein Pixel eine Ausdehnung hat, wohingegen ein Punkt in Geokoordinaten die Ausdehnung 0 hat, d.h. ein Pixel beschreibt einen Bereich in Geokoordinaten; ein Koordinatenintervall wird einem Pixel zugeordnet. Werden nun bei einem Bild die Geokoordinaten des umschließenden Rechtecks angegeben, so stellt sich die Frage der genauen Lage der Pixel dieses Bildes. Unter der Voraussetzung, dass die Geo- und Bildkoordinatenachsen parallel zueinander verlaufen, gibt es im Prinzip vier Möglichkeiten, die Pixel auf die Geokoordinaten zu verteilen (s. Abbildung 4.3). Der Einfachheit halber wird nur eine Dimension des Bildes betrachtet. Das Bild habe in dieser Dimension x_{size} Bildpunkte, die von 0 bis $x_{\text{size}} - 1$ durchnummeriert sind. Der Bereich wird durch die Geokoordinaten x_{gW} (West) und x_{gE} (East) begrenzt.

1. Pixel 0 liegt zentral auf der x_{gW} -, Pixel $x_{\text{size}} - 1$ auf der x_{gE} -Koordinate. Die Pixel werden linear dazwischen verteilt (Abbildung 4.3a). Dies hat den Nachteil, dass die Randpixel auf beiden Seiten je zur Hälfte aus dem mit

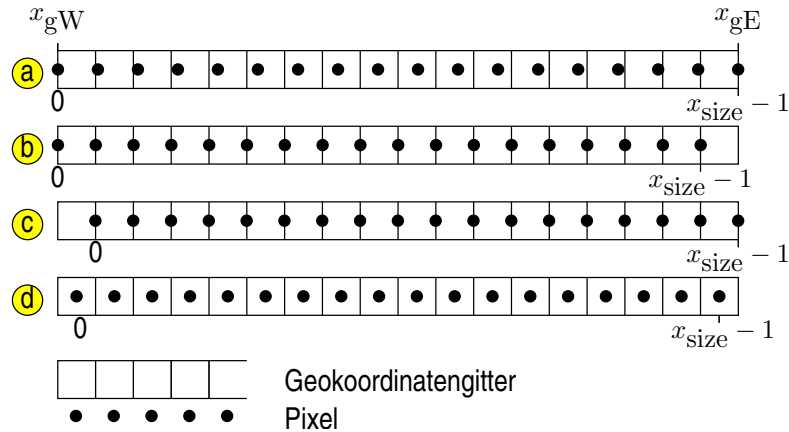


Abbildung 4.3: *Verschiedene Möglichkeiten, Pixel den Geokoordinaten zuzuordnen: a) Pixel 0 und Pixel x_{size} liegen auf den Koordinatengrenzen, b) 0. Pixel liegt auf westlicher Koordinatengrenze x_{gW} , auf der östlichen liegt das Pixel x_{size} ($= 0$. Pixel des nächsten Bildes), c) wie b) nur liegt der Bildausschnitt bündig mit östlicher Koordinate x_{gE} , d) Pixel liegen mittig im Geokoordinatengitter.*

$x_{gW} \dots x_{gE}$ beschriebenen Bereich herausragen. Außerdem ist die Berechnung der Pixelgröße mit $\frac{x_{gE} - x_{gW}}{x_{size} - 1}$ unnötig kompliziert. Das größte Problem ist jedoch, dass bei zwei benachbarten Ausschnitten, die Grenzpixel in beiden Ausschnitten vorkommen.

2. Das letzte genannte Problem tritt bei der in Abbildung 4.3b gezeigten Anordnung nicht auf. Pixel 0 liegt dabei auf der Koordinate x_{gW} . Auf der Koordinate x_{gE} liegt der Pixel x_{size} , der allerdings schon nicht mehr zum Bild dazu gehört. Diese Konstellation hat den Nachteil, dass der Pixel 0 nur zur Hälfte im bezeichneten Gebiet liegt, wohingegen an der östlichen Kante ein halber Pixel fehlt.
3. Die Konstellation Abbildung 4.3c vertauscht gegenüber Abbildung 4.3b lediglich die Randprobleme von West und Ost.
4. Abbildung 4.3d zeigt die optimale Zuordnung von Pixeln und Geokoordinaten. Die Pixel liegen innerhalb des Bereichs $x_{gW} \dots x_{gE}$. Die Pixelgröße

berechnet sich logisch zu $\frac{x_{gE} - x_{gW}}{x_{size}}$. Zwei benachbarte Bereiche haben keinen Überlappungsbereich.

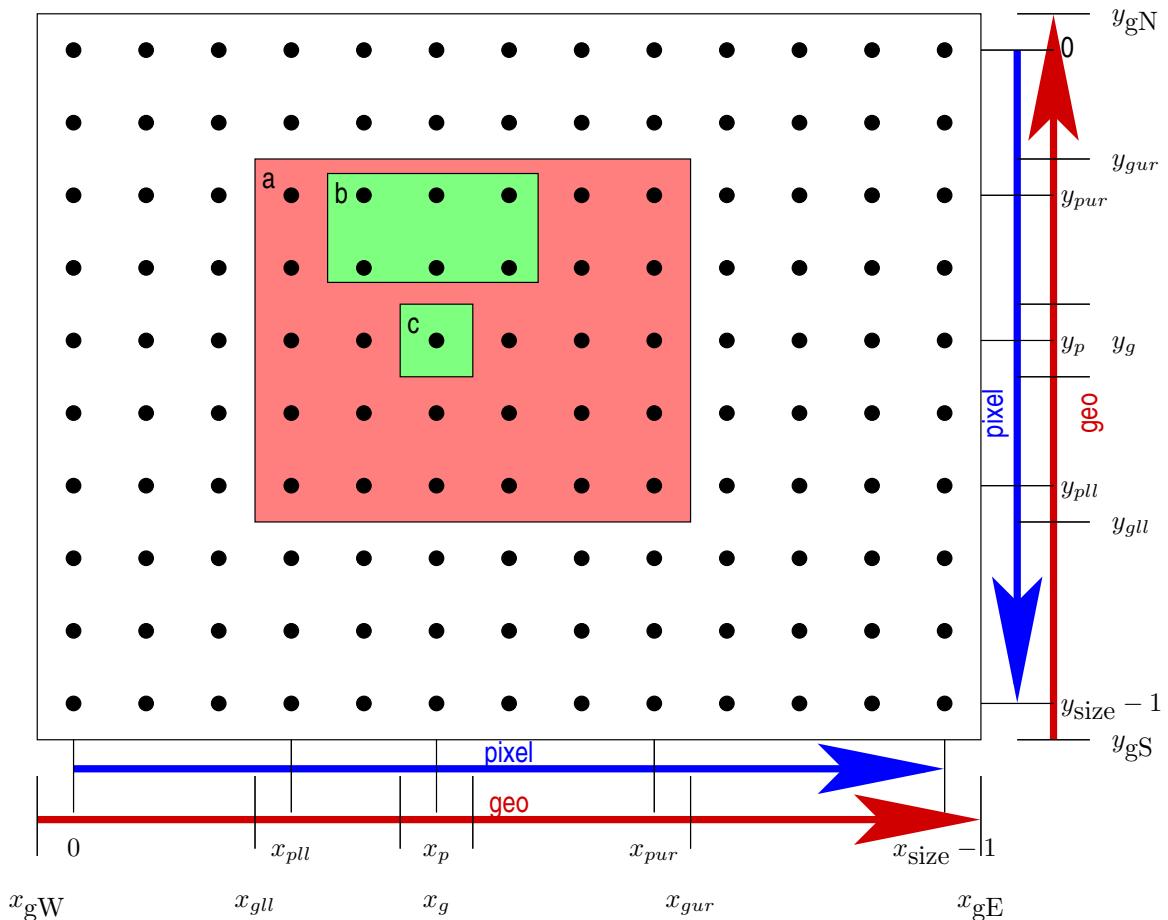


Abbildung 4.4: Umrechnung zwischen Geokoordinaten und Pixeln mit 3 Bereichsbeispielen: a) zeigt die Pixel- und Geokoordinaten-Boundingbox eines Gebiets mit vollständigen Pixeln, b) enthält unvollständige Pixel, c) ist die Region, die einem Pixel zugeordnet wird.

Unter der Bedingung, dass die Pixel vollständig in dem sie beschreibenden Geokoordinatenbereich liegen sollen und dass zwei benachbarte Bildbereiche keine gemeinsamen Bildpunkte haben dürfen, ergibt sich die folgende Umrechnung zwischen Bild- und Geokoordinaten (s. Abbildung 4.4):

$$\begin{aligned}
\frac{x_p - (-0,5)}{x_g - x_{gW}} &= \frac{(x_{\text{size}} - 1 + 0,5) - (-0,5)}{x_{gE} - x_{gW}} \\
\Rightarrow x_p &= \frac{(x_{\text{size}} - 1 + 0,5) - (-0,5)}{x_{gE} - x_{gW}} (x_g - x_{gW}) + (-0,5) \\
&= \frac{x_{\text{size}}}{x_{gE} - x_{gW}} (x_g - x_{gW}) - 0,5 \\
\Rightarrow x_g &= \frac{x_{gE} - x_{gW}}{x_{\text{size}}} (x_p + 0,5) + x_{gW}
\end{aligned}$$

$$\begin{aligned}
\frac{y_p - (-0,5)}{y_g - y_{gN}} &= \frac{(y_{\text{size}} - 1 + 0,5) - (-0,5)}{y_{gS} - y_{gN}} \\
\Rightarrow y_p &= \frac{y_{\text{size}}}{y_{gS} - y_{gN}} (y_g - y_{gN}) - 0,5 \\
\Rightarrow y_g &= \frac{y_{gS} - y_{gN}}{y_{\text{size}}} (y_p + 0,5) + y_{gN}
\end{aligned}$$

In der Tabelle 4.1 sind die Umrechnungsvorschriften zwischen den einzelnen Geo- und Bildkoordinaten sowie zwischen den Boundingboxen zusammengefasst.

4.1.4 Bibliothek zur Verarbeitung von Objektmengen

Bereits in [25] wurde eine Bibliothek mit Funktionen zur Verarbeitung von Objektmengen entwickelt. Diese ermöglicht das Einlesen von Objektmengen, die Bestimmung von Form- und Nachbarschaftsparametern, die dann in eine Bewertung einfließen können, die Gruppierung der Objekte sowie das Ausgeben der Objektgruppen zur Weiterverarbeitung in GEOAIDA.

Die Bibliothek wurde in dieser Arbeit weiterentwickelt und ihr Anwendungsgebiet auf die Klassifizierung von Bildsegmenten ausgedehnt. Insbesondere wurde die parser-gesteuerte, symbolische Verarbeitung von Objektmengen konzeptionell überarbeitet (s. Abschnitt 4.6.2).

$$x_g(x_p) = (x_{gE} - x_{gW})/(x_{size})(x_p + 0,5) + x_{gW} \quad (4.1)$$

$$y_g(y_p) = (y_{gS} - y_{gN})/(y_{size})(y_p + 0,5) + y_{gN} \quad (4.2)$$

$$x_p(x_g) = x_{size}/(x_{gE} - x_{gW})(x_g - x_{gW}) - 0,5 \quad (4.3)$$

$$y_p(y_g) = y_{size}/(y_{gS} - y_{gN})(y_g - y_{gN}) - 0,5 \quad (4.4)$$

$$x_{pll}(x_{gll}) = \text{round}(x_p(x_{gll}) + 0,5) \quad (4.5)$$

$$x_{pur}(x_{gur}) = \text{round}(x_p(x_{gur}) - 0,5) \quad (4.6)$$

$$y_{pll}(y_{gll}) = \text{round}(y_p(y_{gll}) - 0,5) \quad (4.7)$$

$$y_{pur}(y_{gur}) = \text{round}(y_p(y_{gur}) + 0,5) \quad (4.8)$$

$$x_{gll}(x_{pll}) = x_g(x_{pll} - 0,5) \quad (4.9)$$

$$x_{gur}(x_{pur}) = x_g(x_{pur} + 0,5) \quad (4.10)$$

$$y_{gll}(y_{pll}) = y_g(y_{pll} + 0,5) \quad (4.11)$$

$$y_{gur}(y_{pur}) = y_g(y_{pur} - 0,5) \quad (4.12)$$

Tabelle 4.1: Umrechnungsvorschriften für Boundingboxen zwischen Bild- und Geokoordinaten gemäß Abbildung 4.4.

4.1.5 Parser für die Interpretation von XML-Dateien

GEOAIDA besteht aus vielen selbstständigen Komponenten, die Informationen untereinander austauschen müssen. Hierzu sind zwei Dinge zu unterscheiden: das Medium, auf dem Daten ausgetauscht werden, und das Format der Daten. Als Medium stehen beispielsweise Pipes, Sockets oder einfache Dateien zur Verfügung (s. [99]). Pipes stellen eine effiziente Methode für den Datenaustausch zwischen zwei Prozessen dar, die auf demselben Rechner laufen. Sockets ermöglichen den Austausch von Daten über Computergrenzen hinweg. Pipes und Sockets sind gedächtnislos, d.h. wurden die Daten vom lesenden Prozess angefordert, sind sie verloren. Dateien haben den Vorteil, dass sie von allen Programmen zum Einlesen von Daten unterstützt werden. Da GEOAIDA insbesondere auch schon bestehende Programme einbeziehen soll, bietet sich daher diese Form des Datentransfers an.

Das Transfermedium legt nicht das Datenformat fest. Für den Transfer von

Bilddaten wird das in Abschnitt 4.1.1 beschriebene Format verwendet, in das sich Daten leicht wandeln lassen. Benötigt ein Programm ein anderes Format, so muss ein Filter vorgeschaltet werden. Der Austausch von Zusatzinformationen geschieht über ASCII-Dateien, da diese ohne besondere Hilfsmittel im Fall von Störungen lesbar sind. In diesen ASCII-Dateien werden die Informationen in einem an XML angelehnten Format abgelegt, da XML insbesondere für die Beschreibung von attributierten Objekthierarchien entwickelt wurde. Jeder Knoten der Hierarchie wird durch einen Start-Tag eingeleitet und durch einen End-Tag beendet. Zwischen Start- und End-Tag können Subknoten beschrieben werden, die ihrerseits geschachtelt sein können.

Start-Tags werden mit `<` eingeleitet. Unmittelbar auf das `<` folgt der Name des Tags. Dem Namen des Tags kann eine Liste von Attributen folgen, die dem zu beschreibenden Knoten bestimmte Eigenschaften zuordnen. Das Start-Tag wird durch `>` beendet. Enthält der Knoten keine Subknoten, so kann das Start-Tag auch mit `/>` abgeschlossen werden. Es ist dann kein End-Tag mehr erforderlich.

Das **End-Tag** wird mit `</` eingeleitet. Darauf folgt der Name des End-Tags der mit einem zuvor benutzten Start-Tag korrespondieren muss. Ein End-Tag besitzt keine Attribute.

Die beim Start-Tag einsetzbare **Attributliste** hat das Format *Attribute-Name=Value*, wobei mehrere Attribute durch Leerzeichen oder andere Whitespaces getrennt aufeinander folgen können. *Attribute-Name* und *Value* dürfen aus beliebigen Zeichen bestehen. Werden Zeichen verwendet, die auch zum Einleiten oder Beenden der Tags genutzt werden, muss der Ausdruck in Anführungszeichen gesetzt werden. Dem Zeichen " muss ein Backslash \ vorangestellt werden; gleiches gilt natürlich für das Zeichen \ selbst.

Im folgenden Beispiel sind insgesamt drei Knoten dargestellt. Zwei vom Typ `region` und einer vom Typ `subregion`. `subregion` ist Child des `region`-Knoten mit dem Namen `Knoten2`. Der `region`-Knoten mit dem Namen `Knoten1` sowie der `subregion`-Knoten sind einfache Knoten ohne Children. Deshalb können deren Beschreibungen mit `/>` beendet werden, wohingegen der zweite `region`-Knoten ein Endtag (`</region>`) benötigt. Die beiden `region`-Knoten besitzen jeweils die Attribute *id* und *name*; `subregion` besitzt *id* und *label*. Das La-

bel Attribut Baum "Oskar" zeigt dabei, wie Sonderzeichen durch Nutzung des Escape-Zeichen \ in die Attributwerte integriert werden können.

```
<region id=5 name="Knoten1"/>
<region id=3 name=Knoten2>
  <subregion id=6 label="Baum \"Oskar\""/>
</region>
```

4.2 Funktionsweise der Datenanalyse

Die modellgestützte Szenenanalyse bildet den Kern des GEOAIDA-Projekts. In Abbildung 4.5 sind die Komponenten der GEOAIDA-Szenenanalyse dargestellt. Eingangsseitig befinden sich das **semantische Netz**, welches das Modellwissen über die zu interpretierende Szene beschreibt, und die **Datenbasis**, in der die Bild-, Höhen- und GIS-Daten gespeichert sind. Unter GIS-Daten sind hierbei auch Informationen zu verstehen, die bei vorangegangenen Interpretationen ermittelt und in der Datenbank abgelegt wurden. Während der Szeneninterpretation werden die Eingangsdaten mit Hilfe der **Top-Down-** und **Bottom-Up-Operatoren** verarbeitet. **Top-Down-Operatoren** teilen eine Bildregion in Teilregionen und generieren hierzu korrespondierende hypothetische Objekte. **Bottom-Up-Operatoren** bewerten die hypothetischen Objekte und fassen diese zu größeren Einheiten (Gruppen) zusammen. Als Ergebnis der Interpretation erhält man eine **interaktive Karte**, die sich aus der *symbolischen Szenenbeschreibung* und *thematischen Karten* zusammensetzt. Mit Hilfe der interaktiven Karten können dynamisch Karten mit unterschiedlichem Informationsgehalt generiert werden. Der Kern von GEOAIDA ist die Systemsteuerung, welche die Eingangsdaten und das Modellwissen einliest, mit Hilfe der Top-Down-Operatoren Hypothesen für die Szene aufstellt, diese mittels der Bottom-Up-Operatoren bewertet und in Form eines Instanzennetzes und zugehörigen Labelbildern, welche die Lage der Knoten des Instanzennetzes beschreiben, ablegt. Aus diesem Instanzennetz und dem parallel dazu segmentierten Datenmaterial können nun Karten in verschiedenen Detaillierungsgraden generiert werden.

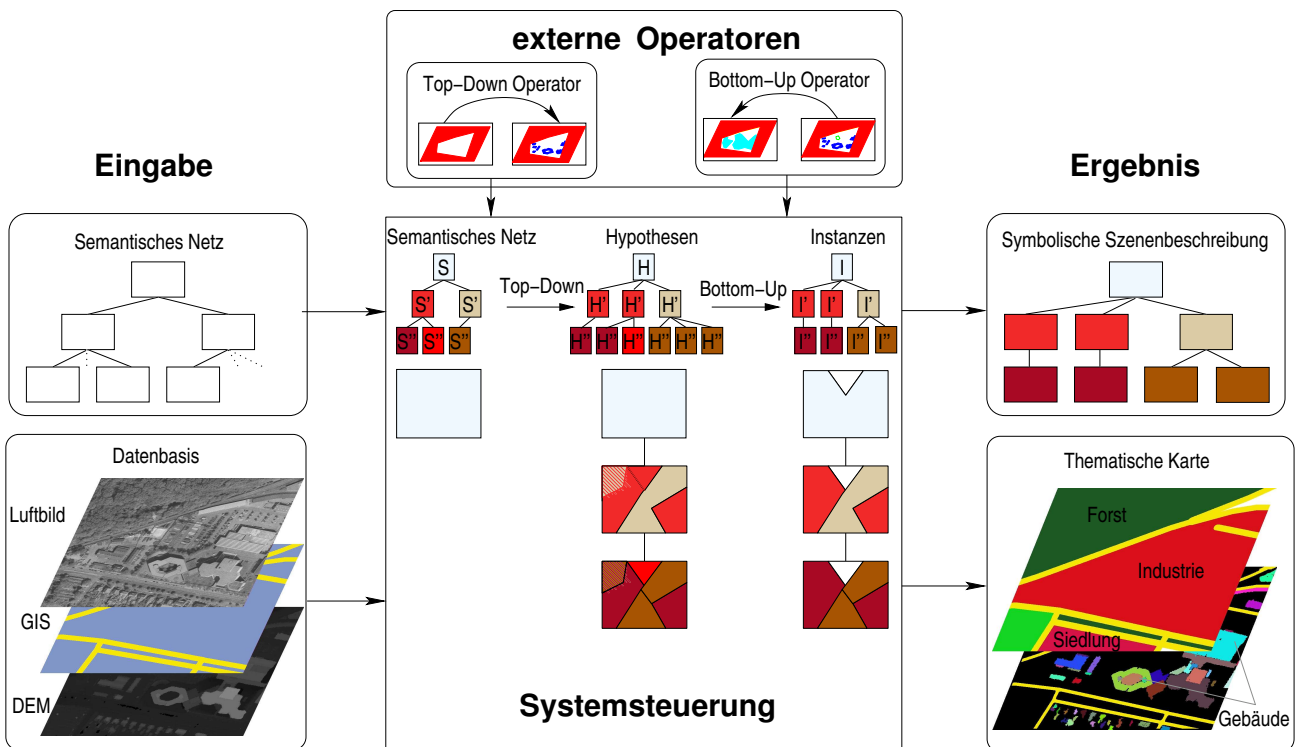


Abbildung 4.5: Aufbau der GEOAIDA-Szenenanalyse: Als Eingangsdaten sind beliebige Fernerkundungsdaten sowie Daten aus einem GIS möglich. Zusammen mit einem Modell der Szene, das in einem semantischen Netz abgelegt ist, erzeugt die Systemsteuerung mit Hilfe externer Operatoren das Ergebnis, das aus einer symbolischen Szenenbeschreibung mit zugehörigen Labelbildern besteht, aus der sich leicht thematische Karten erstellen lassen.

4.2.1 Datenbasis

Unter der Datenbasis sind sämtliche Eingangsdaten zu verstehen, die für die Interpretation der Szene zur Verfügung stehen. Dies können einerseits Bilder unterschiedlicher Sensoren, wie VIS⁵, Laserscan, IR oder SAR sein. Andererseits können aber auch Informationen aus Datenbanken wie GIS oder die Ergebnisse einer früheren Interpretation verwendet werden. Da die eigentliche Anbindung der Daten über die externen Operatoren erfolgt, ergeben sich Einschränkungen nur aufgrund der vorhandenen Bild- und Datenverarbeitungsopere-

⁵visuelles Bild

ratoren. GEOAIDA selbst verwaltet nur zweidimensionale Regionen, die es den Knoten eines Hypothesen- bzw. Instanzennetzes zuweist.

4.2.2 Semantisches Netz

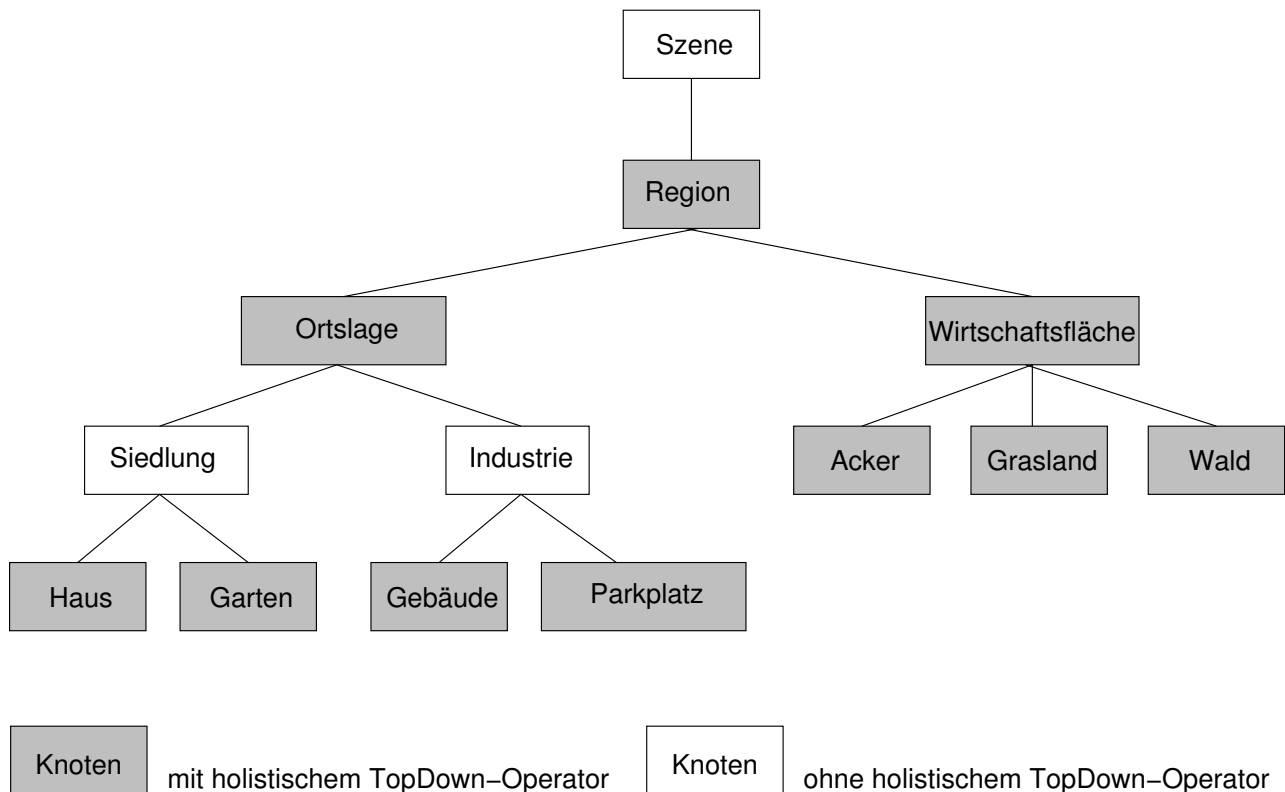


Abbildung 4.6: *Beispiel für ein semantisches Konzeptnetz: Das Konzeptnetz in GEOAIDA ist streng hierarchisch aufgebaut. Bei den Knoten wird unterschieden zwischen Objekten, die per Bildverarbeitungsoperator in den Bilddaten direkt gefunden werden können (holistisch), und solchen die sich nur strukturell, d.h. über deren Komponenten bestimmen lassen. Die Bedeutung der Kanten entspricht durch die hierarchische Struktur generell einer part-of-Relation, wobei dieses die is-a-Relation als Untermenge mit einschließt.*

Das Modellwissen über die zu interpretierende Szene wird in einem semantischen Netz abgelegt (Abbildung 4.6). Die Anordnung der Knoten ist dabei streng hie-

rarchisch, d.h. jeder Knoten hat genau einen übergeordneten Knoten. Der oberste Knoten ist der Szenenknoten. Jeder Knoten entspricht einer in der Szene erwarteten Objektklasse. Jedem Knoten können Attribute zugeordnet werden. Neben allgemeinen Attributen wie *Name* und *Klasse* sind dies zwei Operatoren, *Top-Down* und *Bottom-Up*. Idealerweise ist der *Top-Down*-Operator in der Lage in gegebenem Bildmaterial, Objekte vom Typ der Knotenklasse zu detektieren. Der Knoten besitzt dann einen sog. holistischen Operator. Demgegenüber müssen Knoten ohne holistischen Operator strukturell aus ihren Unterknoten erkannt werden. Aus den detektierten Objekten werden hypothetische Instanzenknoten generiert. Der *Bottom-Up*-Operator untersucht Beziehungen zwischen den Unterknoten und gruppiert sie zu Objekten der Knotenklasse. Aus diesen Objekten werden partielle Instanzenknoten. Die *Top-Down*- und *Bottom-Up*-Operatoren können über weitere Attribute, die operator-spezifisch sind, konfiguriert werden.

Knoten:	Haus
Allgemeine Attribute:	
Name:	Haus
Objectclass:	Haus
Color:	255:0:0
TopDown-Operator:	Detect_Building
TopDown Multi Class:	False
BottomUp-Operator:	None

TopDown-Attribute:	
Input-Data:	Laserscan
Height-Step[m]:	2
Min-Size[m^2]:	30
Max-Size[m^2]:	200

BottomUp-Attribute:	

Abbildung 4.7: *Knoten eines semantischen Netzes mit allgemeinen sowie speziellen Attributen zur Konfiguration der angebundenen Top-Down- und Bottom-Up-Operatoren.*

Abbildung 4.7 zeigt beispielhaft einen Knoten *Haus* eines semantischen Netzes. Das Attribut *Name* bestimmt den Namen unter dem der Knoten im semantischen Netz erscheint. *Class* dient GEOAIDA zur Auswahl der klassifizierten Regionen, nachdem der Top-Down- bzw. Bottom-Up-Operator ausgeführt wur-

de. In der Farbe *Color* werden Objekte diesen Typs in den thematischen Karten dargestellt. Die Attribute Bottom-Up- und Top-Down-Operator legen den Operator fest, der zur Detektion von Objekten dieser Klasse an entsprechender Stelle in der Analyse aktiviert wird. Ist das Attribut *Top-Down-Multi-Class* gesetzt, bedeutet dies, dass der Top-Down-Operator in der Lage ist, mehrere verschiedene Klassen zu detektieren. Mit dieser Art Operator können dann sowohl Objekte der Knotenklasse als auch Objekte von Knoten instanziiert werden, die auf gleicher Ebene den selben übergeordneten Knoten besitzen. Die *Top-Down-Attribute* dienen zur Konfiguration des Top-Down-Operators. Der in dem Beispiel gewählte Top-Down-Operator benötigt beispielsweise als Eingangsdaten ein Höhenbild, einen zu detektierenden Höhengsprung sowie die minimale und maximale Größe der zu erkennenden Regionen. Da es sich beim dargestellten Knoten um einen Endknoten handelt, wird ein Bottom-Up-Operator nicht benötigt.

4.2.3 Top-Down-Operatoren

Top-Down-Operatoren (s. Abbildung 4.8) sind externe Bildverarbeitungsoperatoren, die eine gegebene Menge von Eingangsdaten in Objekte einer oder mehrerer Klassen unterteilen. Zusätzlich erhält der Operator eine Maske, die den zu untersuchenden Bildausschnitt beschreibt. Ist der Bildverarbeitungsoperator nicht in der Lage, Masken zu verarbeiten, so kann der Maskierungsvorgang der Segmentierung auch nachgeschaltet werden. Als Ausgabe erzeugt ein Top-Down-Operator eine Liste von klassifizierten Regionen und ein dazugehöriges Label-Bild, welches die Lage der Regionen beschreibt.

Prinzipiell ist ein Top-Down-Operator nicht auf die Verarbeitung von Bildmaterial beschränkt. Es ist möglich, nur die zu bearbeitenden Geokoordinaten zu übergeben, um hiermit beispielsweise eine Datenbank nach Objekten abzufragen. Zwingend ist lediglich das Ergebnis, das aus einer Regionenliste und dazugehörigen Labelbildern bestehen muss.

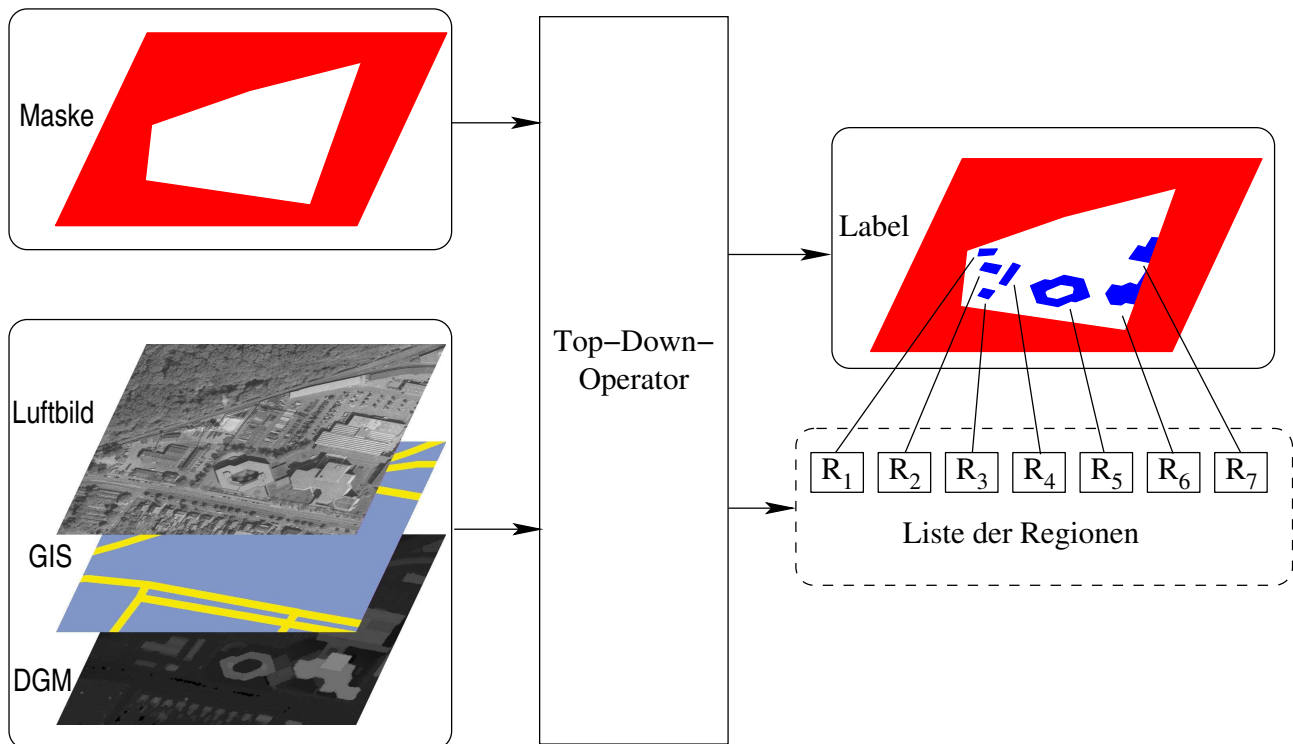


Abbildung 4.8: *Arbeitsweise eines Top-Down-Operators: Der Top-Down-Operator erhält als Eingangsdaten Bildausschnitte der zu bearbeitenden Teilregion. Eine Maske kann diesen Arbeitsbereich weiter einschränken. Als Ausgabe generiert er ein Labelbild und eine zugehörige Liste von klassifizierten Regionen, welche die gefundenen Objekte näher beschreibt.*

4.2.4 Bottom-Up-Operatoren

Die Bewertung und Gruppierung von Objektmengen geschieht in externen Bottom-Up-Operatoren (s. Abbildung 4.9). Diese erhalten eine Liste von Objekten mit dazugehörigen Labelbildern, denen die genaue Lage und Form der Objekte zu entnehmen ist. Mit diesen Informationen und weiteren Attributen, die den Objekten bereits in der Top-Down-Phase zugewiesen wurden, sowie mit Informationen aus externen Quellen wie beispielweise GIS, kann der Operator eine Bewertung durchführen und gegebenenfalls die Form von Objekten anpassen oder Objekte ganz verwerfen. Weitere Eigenschaften, wie Nachbarschaft oder Häufung von Objekten, können in den anschließenden Gruppierungsprozess

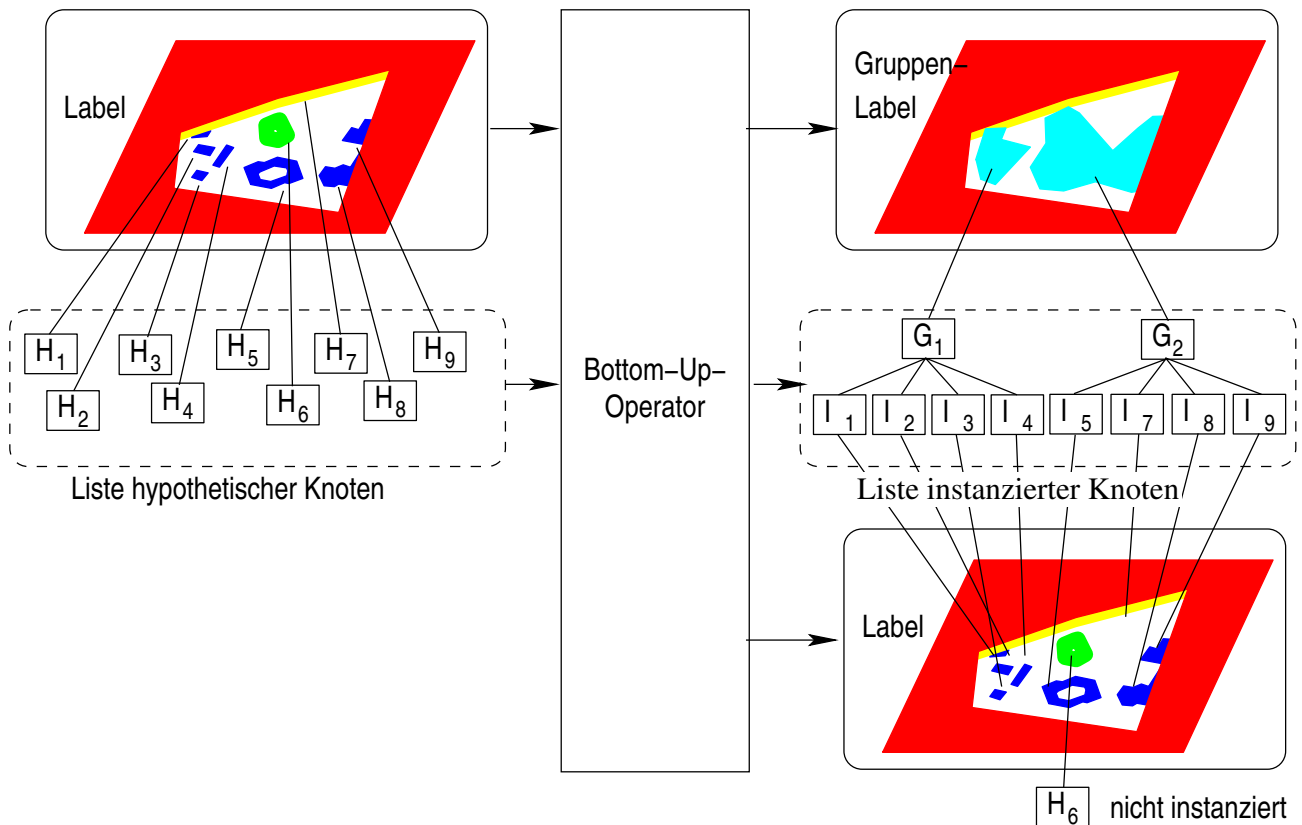


Abbildung 4.9: *Arbeitsweise eines Bottom-Up-Operators: Der Bottom-Up-Operator erhält eine Liste hypothetischer Knoten mit dazugehörigem Labelbild. Diese Knoten werden bewertet, wobei sich die zugehörigen Labelbilder verändern können, und zu übergeordneten Objektklassen gruppiert. Zu den Gruppen wird ein Gruppenlabelbild erzeugt.*

einfließen, in dem Objekte zu übergeordneten Objektklassen zusammengefasst werden. Für diese neuen Gruppenobjekte erstellt der Bottom-Up-Operator ein Labelbild, das deren Lage beschreibt. Zum Abschluss wird die Gruppenbeschreibung zusammen mit den Labelbildner an GEOAIDA übergeben. Innerhalb von GEOAIDA werden die verifizierten Objekte nun zu partiellen Instanzen. Wurde mehr als eine Gruppe erkannt, so wird der ursprüngliche Hypothesenknoten, dessen Child-Knoten im Bottom-Up-Operator ausgewertet wurden, entsprechend geteilt.

4.2.5 Interaktive Karte

Als Ergebnis der Analyse mit GEOAIDA entsteht ein Instanzenetz, in dem sämtliche verifizierte Objekte enthalten sind. Die Anordnung der Knoten ist dabei streng hierarchisch, d.h. der Footprint untergeordneter Knoten ist immer vollständig in dem des übergeordneten enthalten. Zudem sind Knoten einer Hierarchieebene bezüglich ihres Footprints disjunkt. Daher ist es möglich, die Lage sämtlicher Objekte des Instanzenbaumes in einer 2D Karte abzulegen (s. Abschnitt 4.5.1).

Mit Hilfe des semantischen Netzes, des Instanzenetzes und der dazugehörigen Karte ist es möglich, eine interaktive Karte zu realisieren. Das Schließen bzw. Öffnen von Teilbäumen im semantischen Netz bzw. im Instanzenetz führt zur Globalisierung bzw. Detaillierung der Karte bzw. von Kartenausschnitten (s. Abschnitt 4.5.2).

4.2.6 Steuerung der Analyse

Die Kernaufgabe von GEOAIDA ist die Systemsteuerung. Die Analyse läuft dabei in zwei Schritten ab. Zunächst werden im Top-Down-Lauf Hypothesen für in der Szene gefundene Objekte aufgestellt. Zum Generieren der Hypothesen werden holistische Bildverarbeitungsoperatoren eingesetzt. Es entsteht ein Hypothesenetz. Im *Bottom-Up*-Lauf werden die Hypothesen aufgrund von Objekteigenschaften, z.B. Größe, und struktureller Beziehungen zwischen den Objekten, z.B. Nachbarschaftsverhältnissen verifiziert. Es entsteht das Instanzenetz (s. Abschnitt 4.3).

4.3 Ablauf einer Szeneninterpretation

Sei S ein Knoten des semantischen Netzes, H ein hypothetischer Knoten und I ein instanzierter Knoten. N sei ein Knoten im allgemeinen, dann seien N' Unterknoten von N , N'' Unterknoten von N' usw. Mit T_N werde ein Netz von Knoten N bezeichnet, wobei T_N hierarchisch aufgebaut ist, d.h. eigentlich ein

Baum ist.

P beschreibe ein Teilbild, welches dem Knoten N zugeordnet ist. P' ist entsprechend dem Knoten N' zugeordnet. Im Hypothesennetz T_H gelte, alle P' sind Teilmenge von P . Im Instanzennetz gelte zusätzlich: alle Paare von P'_k und P'_l sind disjunkt.

Ziel von GEOAIDA ist es nun das Wissen aus dem semantischen Netz T_S auf einen Bildbereich P anzuwenden und ein Instanzennetz T_I zu erzeugen, welchem das Bildmaterial T_{P_I} zugeordnet ist.

Top-Down-Bearbeitung des semantischen Netzes Die Top-Down-Verarbeitung ist im oberen Teil von Abbildung 4.10 dargestellt. Initial wird dazu aus dem semantischen Knoten S durch Zuordnen des Bildbereichs P der hypothetische Knoten H gebildet.

Der hypothetische Knoten H erzeugt nun mit Hilfe der semantischen Netzknoten S'_i und dem Bildbereich P die hypothetischen Knoten H'_{i_k} , welchen die Bildbereiche $P'_{H_{i_k}}$ zugeordnet werden. Dabei können aus einem S'_i mehrere H'_i vom Typ S'_i entstehen, denen unterschiedliche $P'_{H_{i_k}}$ zugeordnet sind. Ist für den Knoten S'_i kein Top-Down-Operator verfügbar, wird ein einzelner hypothetischer Knoten H'_{i_1} vom Typ S'_i generiert, dem der gesamte Bildbereich P zugeordnet wird. Dieser Vorgang soll als Top-Down-Ausführen von S'_i bezeichnet werden. Besitzt S'_i weitere Unterknoten S'' wird der soeben beschriebene Vorgang für alle H'_{i_k} wiederholt. Aus der Beschreibung der Top-Down-Bearbeitung des semantischen Netzes folgt, dass ein Blattknoten S'' in jedem Fall einen Top-Down-Operator benötigt, da nur so eine Aufteilung der Bilddaten in Teilelemente erreicht werden kann.

Bottom-Up-Auswertung des hypothetischen Netzes Während der *Top-Down*-Bearbeitung wurde das Bildmaterial in Primitive zerlegt. Die *Bottom-Up*-Auswertung (s. Abbildung 4.10 unten) beginnt nun bei den Blattknoten H'' . Dazu startet H' den Bottom-Up-Operator von S' . Dieser bewertet die Knoten H''_i und entscheidet, welche dieser Knoten zu Instanzenknoten I'' werden. Die den Instanzenknoten I'' zugeordneten Bildbereiche P'' sind nun disjunkt. Gleich-

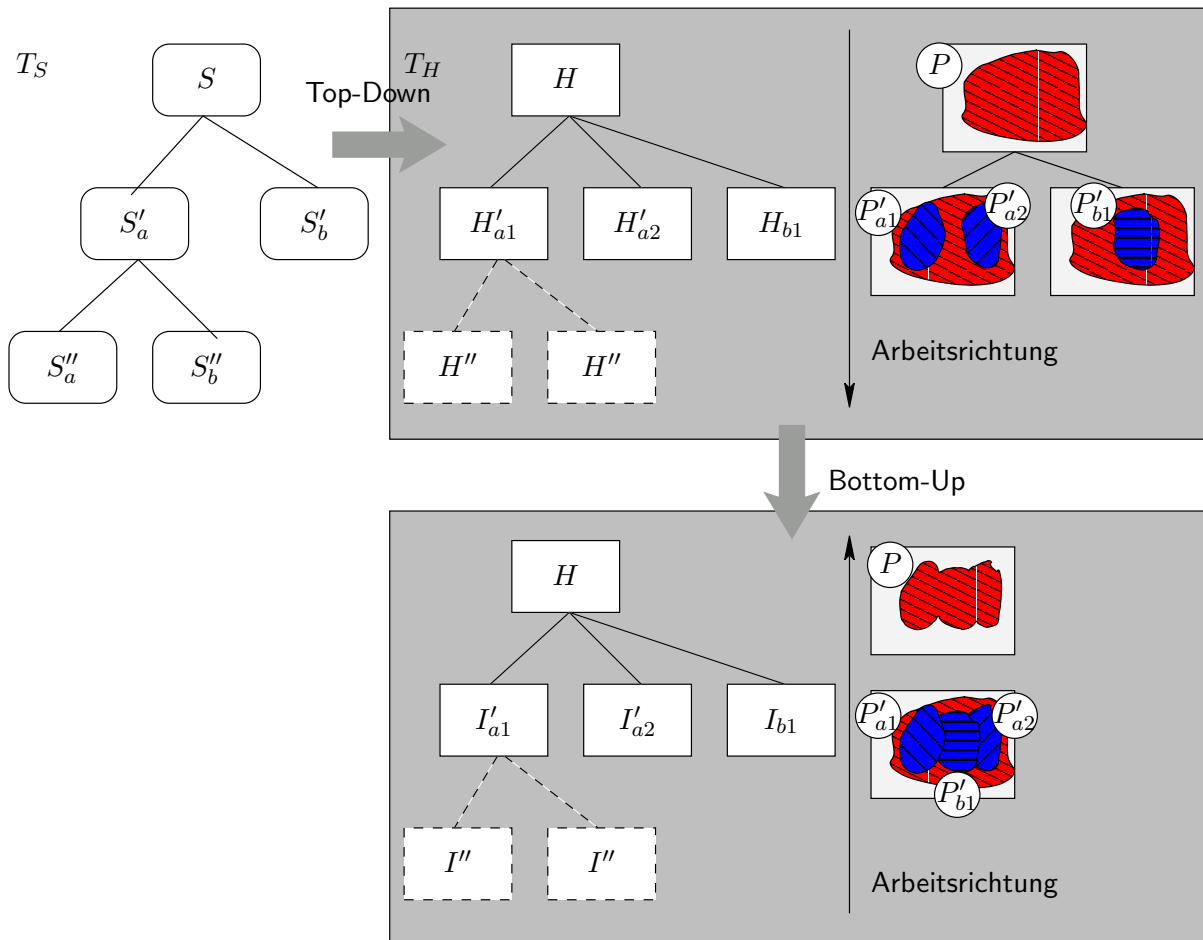


Abbildung 4.10: Ablauf der Szeneninterpretation: Die Top-Down-Phase arbeitet modellgetrieben. Aus dem semantischen Netz T_S wird das Hypothesennetz T_H generiert und über die Top-Down-Operatoren versucht, entsprechende Objekte im Datenmaterial P zu finden. In der Bottom-Up-Phase werden Konflikte zwischen den gefundenen Objekten aufgelöst (datengetrieben), so dass die zugehörigen Regionen P'_i disjunkt sind, und diese zu übergeordneten Objekten gruppiert. Es entsteht ein partielles Instanzennetz.

zeitig gruppiert der Bottom-Up-Operator die neuen Teilinstanzenknoten I'' und generiert neue hypothetische Knoten H'_k , d.h. aus einem Knoten H'_i können bei diesem Schritte mehrere Hypothesen H'_k werden. Die Kontrolle wird nun an den Oberknoten H abgegeben, der die entsprechende Auswertung und Gruppierung für die Knoten H' vornimmt. Dieser Vorgang soll Bottom-Up-Auswertung

von H' bezeichnet werden. Die Bottom-Up-Auswertung wird wiederholt bis der oberste Knoten des hypothetischen Netzes erreicht ist. Mit der Bestätigung des hypothetischen Wurzelknotens werden die darunterliegenden Teilinstanzennetze zu einem Ergebnisnetz instanziiert.

4.3.1 Analysebeispiel

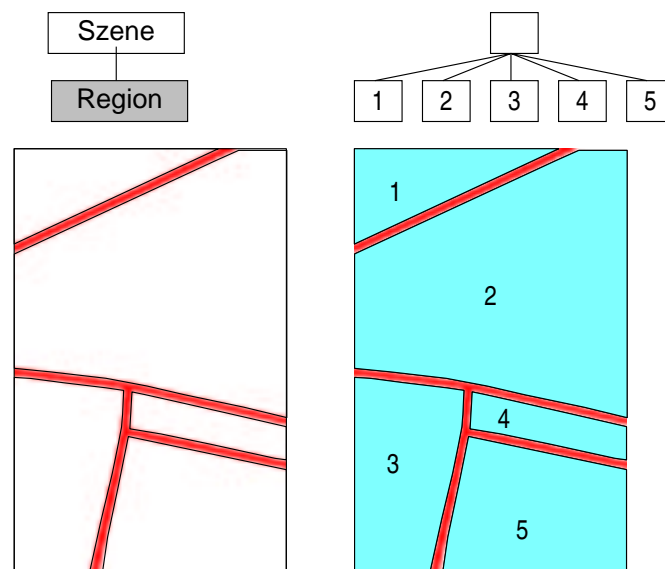


Abbildung 4.11: Beispiel für eine Einteilung des Bildmaterials in von Straßen eingeschlossene Regionen 1...5 mit Hilfe von GIS.

Das folgende Beispiel zeigt wie mit Hilfe von GEOAIDA unter Verwendung von RMK, Laserscan und GIS-Daten eine thematische Karte erstellt werden kann. Als Konzeptnetz dient das in Abbildung 4.6 dargestellte. Ziel ist es, das Bildmaterial (s. Abbildung 4.15, oben) in die Klassen *Industrie*, *Siedlung*, *Wald*, *Grasland* und *Acker* zu zerlegen. Als Segmentierungsoperatoren stehen ein Operator zur Gebäudeerkennung sowie ein Operator zur Texturklassifizierung für *Acker*, *Wald* und *Grasland* zur Verfügung. Im semantischen Netz sind *S*-Knoten, denen ein holistischer Operator zugeordnet ist, grau hinterlegt. Die Klassen *Siedlung* und *Industrie* lassen sich also nur strukturell erkennen.

Die Analyse beginnt mit dem *S*-Knoten *Szene*. Aus dem *S*-Knoten wird ein *H*-Knoten *Szene_1* gebildet. Um Hypothesen seiner Unterknoten zu bilden, werden

die Top-Down-Operatoren aller Unterknoten – in diesem Fall also *Region* – aktiviert. Abbildung 4.11 zeigt das Ergebnis des Top-Down-Operators, der in diesem Beispiel über eine GIS-Abfrage die *Szene_1* in die Regionen 1 bis 5 unterteilt hat. GEOAIDA generiert daraus die *H-Knoten* *Region_1* bis *Region_5*.

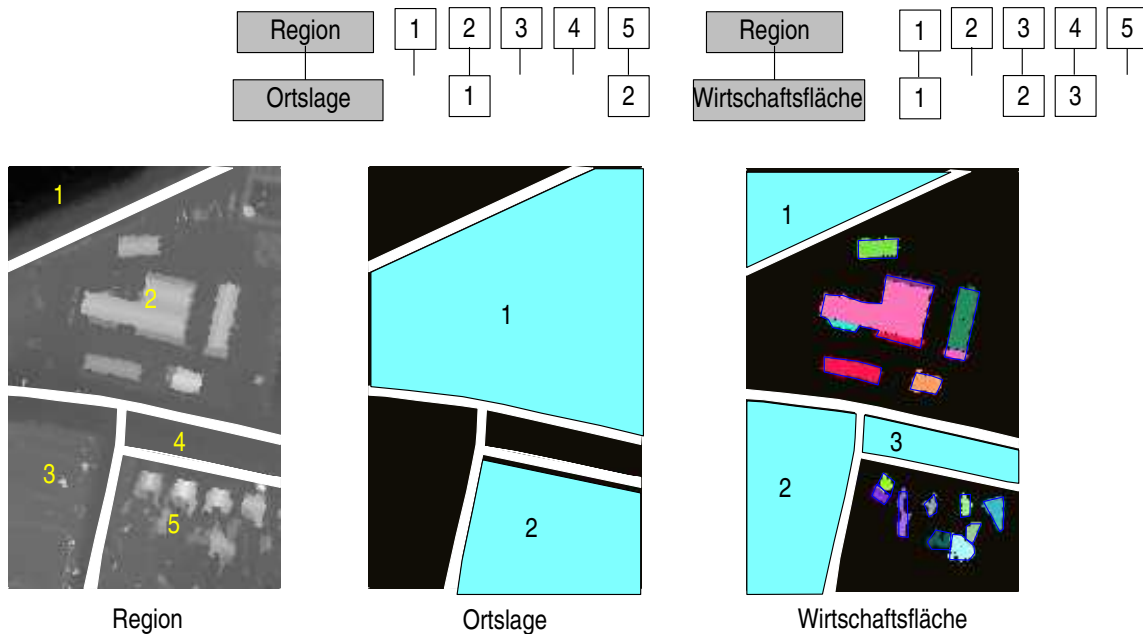


Abbildung 4.12: Beispiel für den binären Einsatz des Algorithmus zur Gebäudeextraktion, um bei den Regionen aus 4.11 zwischen *Ortslage* 1...2 (Gebäude vorhanden) und *Wirtschaftsfläche* 1...3 (ohne Bebauung) zu unterscheiden.

Im nächsten Schritt müssen die *Regionen* 1 bis 5 Hypothesen für ihre Unterknoten bilden. Als Unterknoten können laut Konzeptnetz die Hypothesen *Ortslage* und *Wirtschaftsfläche* entstehen. Jede *Region* ruft daher die Top-Down-Operatoren für *Ortslage* und *Wirtschaftsfläche* auf (s. Abbildung 4.12). Der Top-Down-Operator für *Ortslagen* untersucht, ob in dem betreffenden Gebiet Gebäude vorhanden sind. Ist dies der Fall, wird die *Region* als *Ortslage* klassifiziert. Der Operator für *Wirtschaftsfläche* führt die inverse Funktion aus, d.h. er klassifiziert Gebiete ohne Bebauung als *Wirtschaftsfläche*. Somit werden die *Regionen* 2 und 5 in die *Ortslagen* 1 und 2 überführt, wohingegen die *Regionen* 1, 3 und 4 zu *Wirtschaftsflächen* 1, 2 und 3 werden. GEOAIDA generiert aus den gefundenen *Regionen* die *H-Knoten* *Ortslage_1*, *Ortslage_2* und *Wirtschaftsfläche_1*, *Wirtschaftsfläche_2*, *Wirtschaftsfläche_3*.

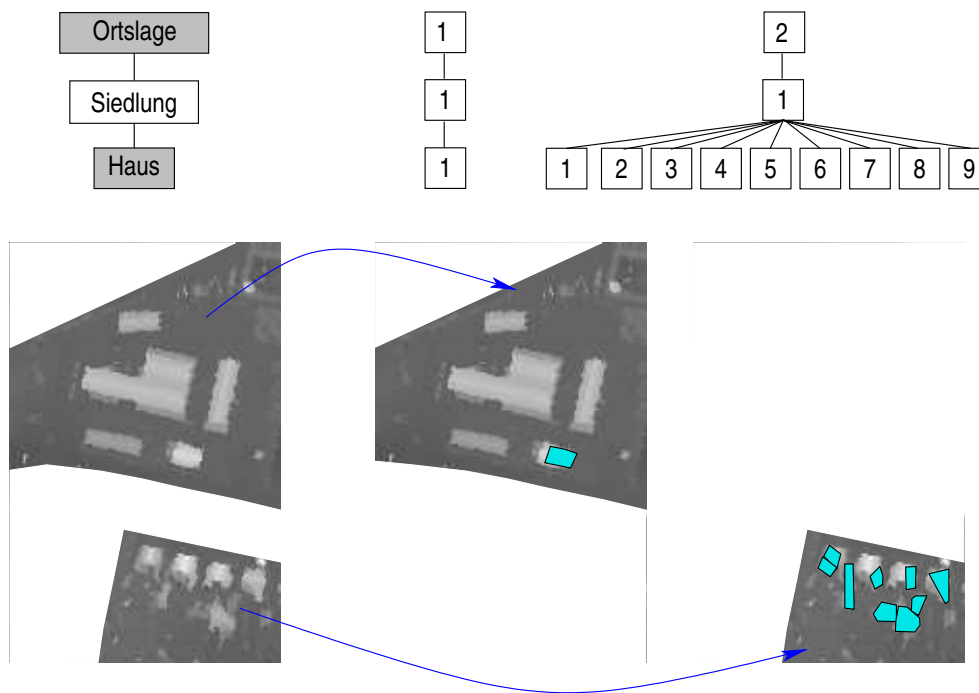


Abbildung 4.13: Beispiel für die Unterscheidung von Siedlungs- und Industriegebieten, durch eine Suche nach kleinen Gebäuden ($30 - 150m^2$).

Im folgenden soll nur noch der Zweig *Ortslage* betrachtet werden. Laut Konzeptnetz kann eine *Ortslage* *Siedlung* oder *Industrie* sein. Da *Siedlung* und *Industrie* keine eigenständigen Top-Down-Operatoren zugeordnet sind, werden zunächst für jede *Ortslage* die Hypothesen *Siedlung* und *Industrie* generiert (s. Abbildung 4.13 und 4.14). Eine *Siedlung* besteht aus *Häusern* und *Gärten*. Daher rufen die *H*-Knoten *Siedlung* die Top-Down-Operatoren für *Haus* und *Garten* auf. In Abbildung 4.13 ist das Ergebnis der *Haus*-Segmentierung exemplarisch dargestellt. Als Operator fungiert der schon zur Unterscheidung in *Ortslage* und *Wirtschaftsfläche* eingesetzte Gebäudesegmentierer in Laserscandaten. Parametrisch ist er hier auf die Detektion von Gebäuden (Häuser) mit Grundflächen zwischen $30m^2$ und $150m^2$ eingestellt. In der Region *Ortslage_1* wird lediglich ein passendes Haus detektiert. In der Region *Ortslage_2* sind es 9.

Der gleiche Operator wird – parametrisch auf die Detektion von Gebäuden mit Grundflächen zwischen $150m^2$ und $500m^2$ eingestellt – für die Erkennung von wirtschaftlich genutzten Gebäuden eingesetzt (s. Abbildung 4.14). In der

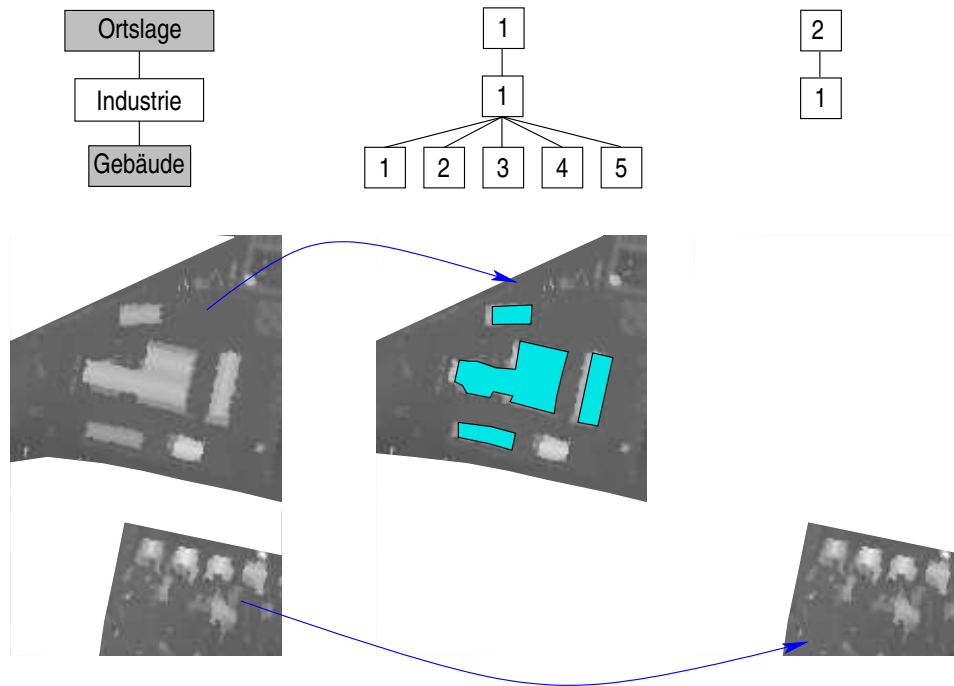


Abbildung 4.14: *Beispiel für die Erkennung von wirtschaftlich genutzten Gebäuden durch eine Suche nach Objekten der Größe 150 – 500m².*

Ortslage₁ zugeordneten Region werden 5 Gebäude mit dieser Eigenschaft gefunden, in der Ortslage₂ zugeordneten keine.

Ist die Segmentierung an den Blättern des Konzeptnetzes angekommen, so beginnt an dieser Stelle nun die Bewertung. *Häuser* und *Gärten* werden zu *Siedlungen* zusammengefasst, *Gebäude* und *Parkplätze* zu *Industrie*. Bei der Fusion von *Siedlungen* und *Industrie* tritt in Ortslage₁ ein Konflikt zwischen den beiden auf. Dieser wird zugunsten von *Industrie* aufgelöst, da *Industrie* den größeren Flächenanteil stellt. Denkbar wäre hier natürlich auch eine Aufspaltung in zwei Teilregionen. Die Fusion wird fortgesetzt bis der Knoten *Szene₁* erreicht ist. Alle erkannten *H*-Knoten werden während dieser Bottom-Up-Evaluation in Instanzen (*I*-Knoten) umgewandelt. Mit Hilfe des entstandenen Instanzenbaums und den zugehörigen Regionen im Bildmaterial lassen sich schließlich thematische Karten generieren. Abbildung 4.15 zeigt eine solche Karte, in der neben der Landnutzung auch noch die erkannten Gebäude eingetragen sind.

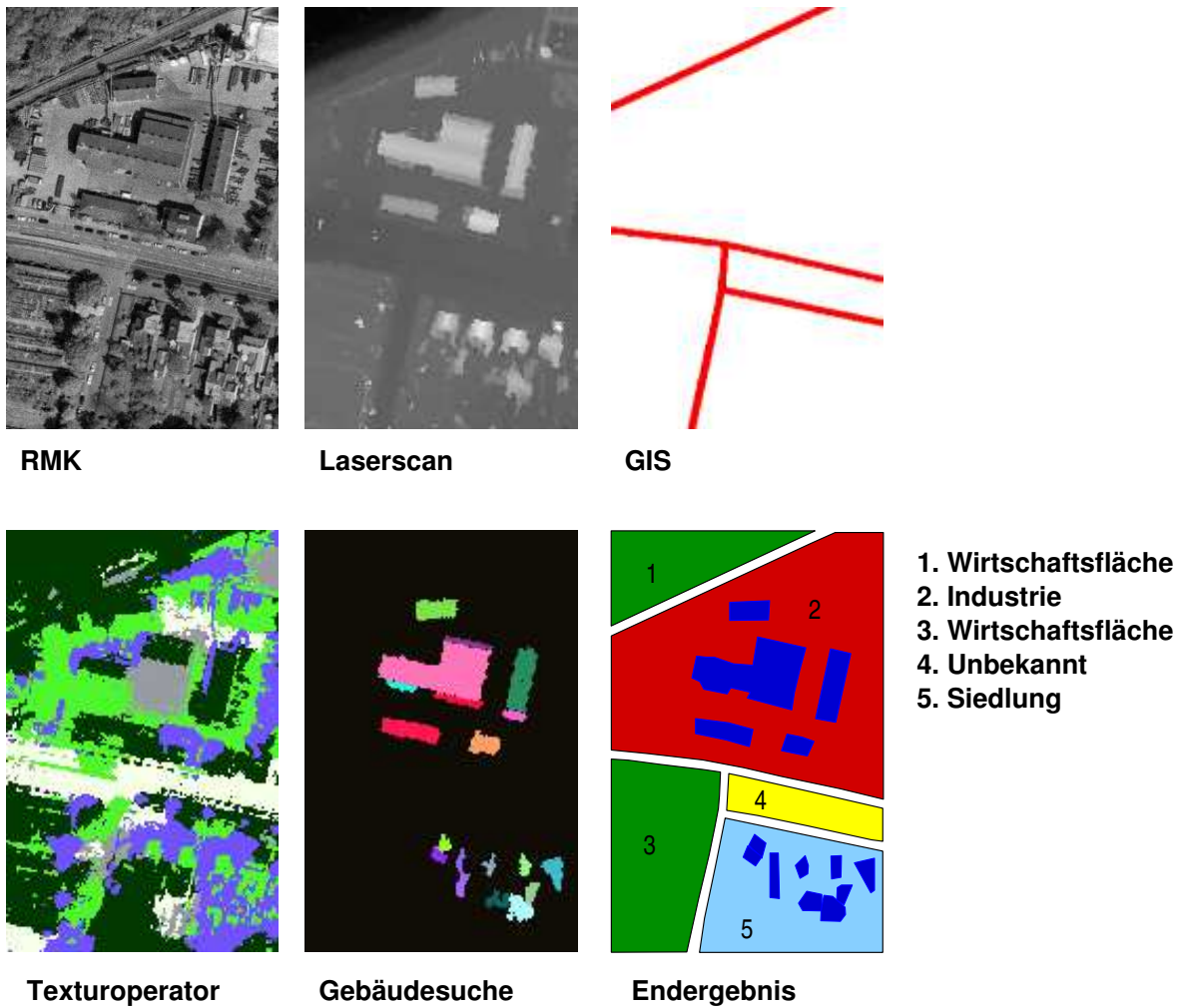


Abbildung 4.15: Übersicht über die Ein- und Ausgangsbilder für eine Szeneninterpretation: oben) Eingangsbilder (RMK, Laserscan, GIS), unten) Segmentierungsergebnisse (Texturoperator, Gebäudesuche), Endergebnis.

4.4 Technische Realisierung

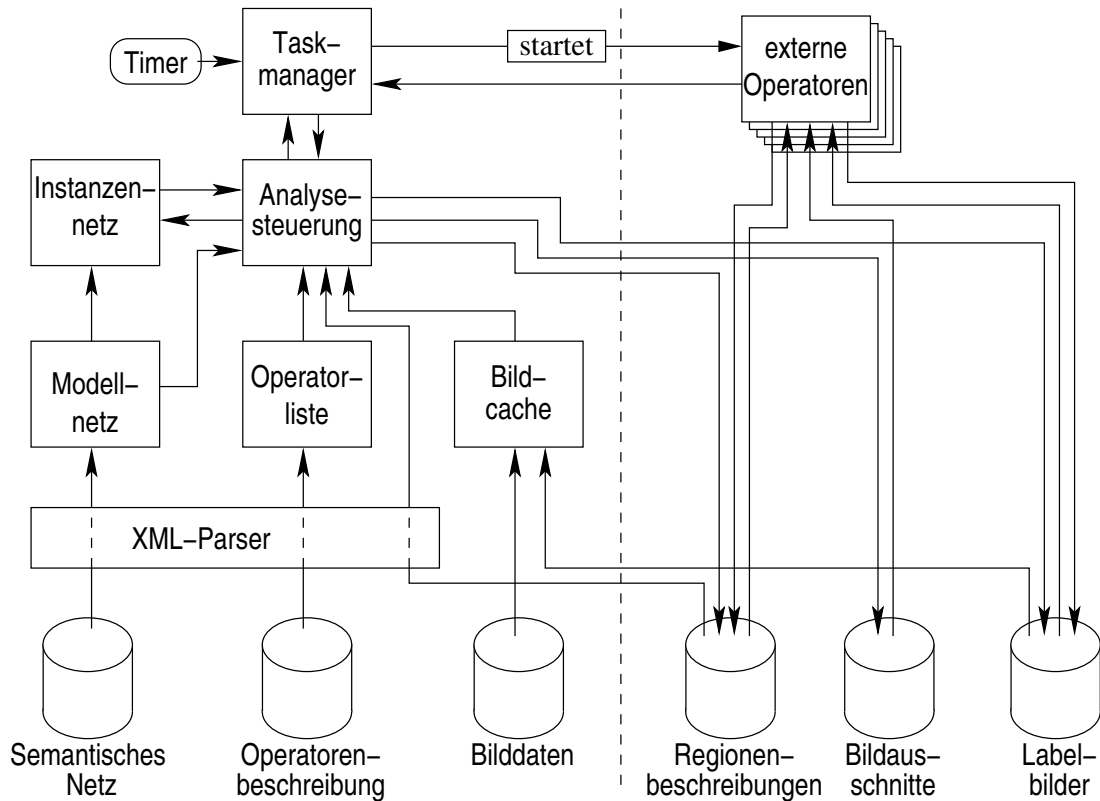


Abbildung 4.16: Schematische Funktionsübersicht über GEOAIDA.

Die technische Umsetzung von GEOAIDA ist schematisch in Abbildung 4.16 dargestellt. Dabei bleibt das grafische Benutzerinterface, zu dem auch der Netzeditor sowie die Visualisierungseinheit gehören, unberücksichtigt. Um die Effizienz des Gesamtsystems zu steigern, werden Techniken wie Parallelisierung der Verarbeitung und Caching-Mechanismen bei der Bildverwaltung angewendet. Im Mittelpunkt steht die **Analysesteuerung**. Diese verwaltet das **Instanzennetz**, für dessen initialen Aufbau sie Informationen aus dem **Modellnetz** (semantisches Netz) erhält. Für die Entwicklung des Instanzennetzes werden Informationen über externe Operatoren angefordert. Die Ausführung der **externen Operatoren** wird vom **Taskmanager** kontrolliert, der dafür sorgt, dass der Rechner optimal ausgelastet wird, indem er **timer**-gesteuert die Recherauslastung ermittelt. Informationen über Aufrufmodalitäten erhält die Analyse aus dem Modul **Operatorliste**. Dieses Modul wird ebenso wie das Modell-

netz vor dem Start der Analyse aus einer Datei initialisiert, die es mit Hilfe des **XML-Parsers** interpretiert. Die effiziente Speicherausnutzung wird durch einen **Bild-Cache** unterstützt. Sämtliche Bilddaten werden über den Cache geladen, der sowohl eine mehrfache Datenhaltung verhindert als auch länger nicht benötigte Bilddaten aus dem Speicher entfernen kann. Der Datenaustausch mit den externen Operatoren geschieht über **Regionenbeschreibungen**, die ebenfalls durch den **XML-Parser** geleitet werden, sowie über Labelbilder. Für die Top-Down-Operatoren werden zusätzlich Bilddateien erzeugt, welche die Daten für den aktuell zu bearbeitenden Ausschnitt enthalten.

4.4.1 Taskmanager

GEOAIDA ist in der Lage, viele Prozesse parallel anzustoßen. Werden die Prozesse tatsächlich alle gleichzeitig gestartet, kann dies jedoch schnell zu einer Überlastung des Rechners führen. Deshalb startet die Analyse externe Programme nicht direkt, sondern übergibt sie dem Taskmanager. Der Taskmanager enthält eine Warteschlange (Queue). Wird die Ausführung eines Programmes beantragt, wird dessen Befehlszeile zunächst zusammen mit dem aufrufenden Objekt in die Warte-Queue gestellt. Per Timer wird ständig die Rechnerauslastung überwacht. Sobald die Rechnerauslastung es zulässt, wird das Programm gestartet und aus der Warte-Queue in einen Datencontainer mit den laufenden Programmen übertragen. Am Programmende angelangt, erhält der Taskmanager ein Signal, woraufhin er dem Objekt, das die Programmausführung veranlasst hat, diesen Zustand signalisiert. Das betreffende Objekt kann dann seine Arbeit fortsetzen.

4.4.2 Analysesteuerung

Die Analysesteuerung ist rekursiv über eine in jedem Instanzenknoten enthaltene FSM⁶ realisiert. Durch Implementierung mittels FSM ist die Analyse parallelisierbar, da jede zu durchlaufende Phase unabhängig von externen Ereignissen ist. Externe Ereignisse können z.B. die Ausführung eines Operators oder

⁶Finite State Machine

die Generierung eines untergeordneten Instanzennetzes sein. Die FSM kann die Zustände `TD_START`, `TD_EVALUATE`, `START_CHILDREN`, `BU_START` und `BU_EVALUATE` annehmen und durchläuft diese im Normalfall auch in dieser Reihenfolge. Hinzu kommen Fehlerzustände, die aber der Übersicht halber nicht näher betrachtet werden sollen. In der Phase `TD_START` wird der Top-Down-Operator gestartet, Phase `TD_EVALUATE` wertet die Ergebnisse des Top-Down-Operators aus. In der nächsten Phase `START_CHILDREN` werden Starthypothesen für die Unterknoten der in der letzten Phase aus dem Ergebnis des Top-Down-Operator-Laufs neu generierten Hypothesen erzeugt und die FSM für die Unterknoten in Gang gesetzt. Es folgt die `BU_START`-Phase, in dem die Evaluation der Hypothesen durch Starten der Bottom-Up-Operatoren eingeleitet wird. In der letzten Phase `BU_EVALUATE` werden die Ergebnisse dieser Evaluation schließlich ausgewertet.

Die in jeder Phase auszuführenden Programmschritte sind in den Abbildungen 4.17, 4.18, 4.19, 4.20 und 4.21 in einer Pseudoprogrammiersprache dargestellt. In den Programmtexten bezeichnet `Inode` die Klasse eines instanziierten Knotens. `snode` ist der zum aktuellen Instanzenknoten gehörige Modellknoten des semantischen Netzes. `state` speichert die Phase, in welche die FSM beim nächsten Aufruf übergehen soll.

START_CHILDREN-Phase Zunächst wird der Szenenknoten des semantischen Netzes instanziiert. Die FSM dieses Knotens hat den Zustand `START_CHILDREN`. Der Aufruf der knotenspezifischen FSM bewirkt den in Abbildung 4.17 dargestellten Programmablauf. Zunächst wird ein Zähler `count`, der die Zahl der in Verarbeitung befindlichen Subknoten zählt, initialisiert. Dann wird für jeden Subknoten des semantischen Netzes `count` inkrementiert, ein hypothetischer Instanzenknoten `hnode` angelegt und in den Instanzenbaum eingebunden sowie der Suchbereich für den Top-Down-Operator des neu angelegten Knotens – hier angedeutet durch die Zuweisung der Boundingbox – festgelegt. Der Zustand der FSM `hnode.state` wird auf `TD_START` gesetzt und die FSM des neuen Knotens gestartet. Der Zustand der FSM wird nun für den nächsten Aufruf auf `BU_START` gesetzt.

```
procedure INode.fsm()
begin
  case state in
  begin
  START_CHILDREN:
    count:=0;
    for child in snode.children() do
    begin
      count:=count+1;
      hnode=new INode(child);
      childLink(hnode);
      hnode.geoBBox=geoBBox;
      hnode.state:=TD_START;
      hnode.fsm();
    end;
    state:=BU_START;
  ...
  end;
end;
```

Abbildung 4.17: FSM: Phase START_CHILDREN

TD_START-Phase In der TD_START-Phase (s. Abbildung 4.18) werden die Befehlszeile für den Top-Down-Operator generiert sowie die Bildausschnitte erstellt, die für die Ausführung des Top-Down-Operators benötigt werden. Der Zustand der FSM wird auf TD_EVALUATE gesetzt. Schließlich wird die auszuführende Befehlszeile zusammen mit dem Objekt, das diese Operation ausführen möchte, beim Taskmanager registriert. Dieser bringt das gewünschte Programm zur Ausführung, sobald genügend Rechenkapazität zur Verfügung steht. Nach Beendigung des externen Programms ruft der Taskmanager die FSM des anfordernden Hypothesenknotens auf, die sich daraufhin in der TD_EVALUATE-Phase befindet.

TD_EVALUATE-Phase Aufgabe der TD_EVALUATE-Phase ist die Auswertung der Ergebnisse des Top-Down-Operators. Dazu wird zunächst die vom Top-Down-Operator geschriebene Regionenliste mit Hilfe des XML-Parsers eingelesen und aus diesen Beschreibungen eine Liste von hypothetischen Instanzenkno-

```
procedure INode.fsm()
begin
  case state in
  begin
    ...
    TD_START:
      command:=snode.generateTopDownCommand();
      snode.generateImages(geoBBox);
      state:=TD_EVALUATE;
      task.register(this,command);
    ...
  end;
end;
```

Abbildung 4.18: FSM: Phase TD_START

ten generiert. Diese neuen Knoten werden nun in das Instanzennetz eingehängt, sowie der Zähler für die zu bearbeitenden Unterknoten des übergeordneten Knotens inkrementiert. Der Zustand der FSM der neuen Knoten wird auf `START_CHILDREN` gesetzt. Besitzt der zugehörige semantische Netzknoten Unterknoten, so werden für diese durch Start der Knoten FSM `hnode.fsm()` nun Hypothesen angelegt. Ist der hypothetische Knoten dagegen ein Blattknoten, so wird die FSM des übergeordneten Knotens gestartet. Nachdem die Liste der neuen Knoten abgearbeitet wurde, wird die Hypothese selbst aus dem Instanzennetz entfernt, die FSM des übergeordneten Knotens aufgerufen und die Hypothese schließlich gelöscht.

BU_START-Phase Die `BU_START`-Phase dient zum Starten der Bottom-Up-Operatoren. Zunächst wird die Anzahl der in Arbeit befindlichen Unterknoten dekrementiert. Sind alle Unterknoten bearbeitet worden, so wird die Knotenliste generiert, die der Bottom-Up-Operator bearbeiten soll, und die Befehlszeile wird generiert. Nachdem der Zustand der FSM für die nächste Phase `BU_EVALUATE` vorbereitet wurde, wird das auszuführende Kommando beim Taskmanager registriert. Dieser startet die FSM erneut, sobald das Kommando ausgeführt und beendet wurde.

```
procedure INode.fsm()
begin
  case state in
  begin
    ...
  TD_EVALUATE:
    hodelist:=snode.readTopDownResult();
    for hnode in hodelist do
      begin
        parent.childLink(hnode);
        parent.count:=parent.count+1;
        hnode.state=START_CHILDREN;
        if (hnode.snode.numberOfChildren==0)
        then parent.fsm();
        else hnode.fsm();
      end;
    parent.childUnlink(this);
    parent.fsm();
    delete this;
    ...
  end;
end;
```

Abbildung 4.19: FSM: Phase TD_EVALUATE

BU_EVALUATE-Phase Die BU_EVALUATE-Phase ist die letzte Phase. Sie wird vom Taskmanager getriggert. In ihr werden die Resultate des Bottom-Up-Operators ausgewertet. Dazu wird zunächst das vom Bottom-Up-Operator generierte Knotennetz mit Hilfe des XML-Parser eingelesen. Handelt es sich beim aktuellen Knoten um den Wurzelknoten, also den Szenenknoten, so ist die Analyse beendet und die interaktive Karte kann generiert werden. Ansonsten werden die neuen Gruppenknoten beim übergeordneten Knoten eingehängt. Der aktuelle Knoten wird aus dem Instanzennetz entfernt und die FSM des übergeordneten Knotens wird gestartet. Zum Schluss wird der aktuelle Knoten gelöscht, da es sich nur um eine Hypothese handelte.

```
procedure INode.fsm()
begin
  case state in
  begin
    ...
    BU_START:
      count:=count-1;
      if count=0
      then
        begin
          generateNodeList(this.children);
          command:=snode.generateBottomUpCommand();
          state:=BU_EVALUATE;
          task.register(this,command);
        end;
      ...
    end;
  end;
end;
```

Abbildung 4.20: FSM: Phase BU_START

4.4.3 Informationsaustausch mit externen Komponenten

Die eigentliche Bildverarbeitungsfunktionalität wird in GEOAIDA über sogenannte Operatoren eingebunden. Dies sind eigenständige Programme, die über ein definiertes Interface mit GEOAIDA kommunizieren.

Der Informationsaustausch zwischen den einzelnen Komponenten von GEOAIDA geschieht über Kommandozeilenparameter und ASCII-Dateien, in denen die Daten in einem XML-angelehnten Format gespeichert werden. Kommandozeilenparameter erlauben die einfache Übergabe von Konfigurationsparametern an externe Programme. Diese Methode kommt beim Aufruf der Top-Down- und Bottom-Up-Operatoren zur Anwendung. Komplexe Objekthierarchien, wie sie beispielsweise von den Top-Down- bzw. zu und von den Bottom-Up-Operatoren übertragen werden, lassen sich dagegen sehr einfach in XML darstellen. Jeder Knoten der Hierarchie wird durch einen Start-Tag eingeleitet und durch einen End-Tag beendet. Zwischen Start- und End-Tag können Subknoten beschrieben werden, die ihrerseits geschachtelt sein können.


```
procedure INode.fsm()
begin
  case state in
  begin
    ...
    BU_EVALUATE:
      godelist:=snode.readBottomUpResult();
      if isRoot
      then analysisFinished(godelist.first);
      else
        begin
          for gnode in godelist do
            begin
              parent.childLink(gnode);
            end;
            parent.childUnlink(this);
            parent.fsm();
          end;
          delete this;
        end;
      end;
end;
```

Abbildung 4.21: FSM: Phase BU_EVALUATE

4.4.3.1 Bilddatenbeschreibung

Die Bilddaten, der in GEOAIDA verwendeten Eingangsdaten werden mit dem Tag `geoimage` beschrieben. Die folgende Attribute müssen gesetzt werden:

file ist der Dateiname der zu diesem Bild gehörenden Rasterdaten.

geoType bestimmt die Art des Geokoordinatensystems. Bisher wird lediglich GK/1, GK/2 und GK/3 (Gauß-Krüger) unterstützt. Denkbar wären jedoch auch Typen wie UTM.

geoNorth, geoSouth, geoWest, geoEast beschreibt die Geokoordinaten der Rasterdaten im mit `geoType` festgelegten Koordinatensystem. Die Geokoordinaten beschreiben die Bildpunkte $(-0.5, size_y-0.5)$ und $(size_x-0.5, -0.5)$ (s. auch 4.1.3 auf Seite 45).

type bestimmt den Typ dieses Bildes. Derzeit sind die Bildtypen **LASER**, **VIS**, **SAR**, **IR** und **VIDEO** möglich. Im Prinzip ist jeder beliebige Bildtyp möglich, sofern in der Datenbank ebenfalls Bilder diesen Typs für die angewählte Region existieren.

key dient dazu, das Bild innerhalb seiner **type**-Klasse eindeutig zu identifizieren.

name ist der Name des Bildes.

4.4.3.2 Regionenbeschreibung

Eine Regionenbeschreibung entsteht als Ergebnis eines Top-Down-Operators. Sie besteht aus einer Liste von Tags des Typs **region**. Die folgende Attribute müssen ggf. gesetzt werden:

class bestimmt die Klasse, von welcher diese Region ist. Nur Regionen, die der Klasse des aufrufenden Knotens entsprechen, werden in das Instanzennetz überführt.

id: Unter der angegebenen *id* ist die Region im Labelbild zu finden.

file ist der Dateiname des zu dieser Region gehörenden Labelbildes. Mehrere Regionen können das gleiche Labelbild verwenden. Die Unterscheidung findet über die *id* statt.

llx, **lly**, **urx**, **ury** beschreiben die Boundingbox der Region in Pixeln im angegebenen Labelbild *file*. Dabei gilt: $llx \leq x \leq urx$ und $ury \leq y \leq lly$, d.h. die Boundingbox liegt vollständig in der Region.

geoNorth, **geoSouth**, **geoWest**, **geoEast** beschreibt die Boundingbox der Region. Die beschriebene Boundingbox in Geokoordinaten liegt genau auf der Regionengrenze und ist damit in jeder Richtung einen halben Pixel größer als die Boundingbox in Pixeln. Geokoordinaten müssen nicht angegeben werden. Dann wird jedoch angenommen, dass die Geokoordinaten und die Größe des Gesamtbildes (**file**) denen der übergeordneten Region entsprechen.

name ist der Name der Region. Er dient lediglich Visualisierungszwecken. Ist kein *name* angegeben, wird ein Name automatisch aus dem Klassennamen und einer fortlaufenden Nummer generiert.

4.4.3.3 Knotenliste

An einen Bottom-Up-Operator wird eine Knotenliste übergeben, dessen Aufgabe darin besteht:

- die Knoten zu gruppieren,
- ein Labelbild der Knoten zu generieren,
- ein Labelbild der Gruppenknoten zu erzeugen.

Knoten werden mittels `<node` eingeleitet, Gruppenknoten mittels `<group`.

Als Eingabe erhält der Bottom-Up-Operator eine Liste von nodes:

```
<node name=... file=... id=... addr=... .../>
<node name=... file=... id=... addr=... .../>
<node name=... file=... id=... addr=... .../>
...
```

Hieraus erzeugt der Bottom-Up-Operator eine gruppierte Liste der Form:

```
<group id=... file=... llx=... ...>
  <node name=... file=... id=... addr=... .../>
  <node name=... file=... id=... addr=... .../>
  <node name=... file=... id=... addr=... .../>
  ...
</group>
<group id=... file=... llx=... ...>
  ...
</group>
...
```

`nodes` und `groups` können beliebige Attribute haben. Die folgenden Attribute sind in den `nodes` der Eingabedatei in jedem Fall vorhanden.

class bestimmt die Klasse, von welcher diese Region ist.

id: Unter der angegebenen *id* ist die Region im Labelbild zu finden.

file ist der Dateiname des zu dieser Region gehörenden Labelbildes. Mehrere Regionen können das gleiche Labelbild verwenden. Die Unterscheidung findet über die *id* statt.

llx, lly, urx, ury beschreiben die Boundingbox der Region in Pixeln im angegebenen Labelbild *file*. Dabei gilt: $llx \leq x \leq urx$ und $ury \leq y \leq lly$, d.h. die Boundingbox liegt vollständig in der Region.

geoNorth, geoSouth, geoWest, geoEast beschreibt die Boundingbox der Region. Die beschriebene Boundingbox in Geokoordinaten liegt genau auf der Regionengrenze und ist damit in jede Richtung einen halben Pixel größer als die Boundingbox in Pixeln.

file_geoNorth, file_geoSouth, file_geoWest, file_geoEast beschreibt die Boundingbox des mittels `file` angegebenen Label-Bildes.

name ist der Name des Knotens.

addr gibt die Adresse dieses Attributs im Speicher an. Dieses Attribut darf in keinem Fall verändert werden, da es die Zuordnung dieses Knotens im Instanzennetz herstellt.

In den einzelnen Knoten muss der Bottom-Up-Operator für Knoten, die übernommen wurden, das Attribut *addr* und gegebenenfalls veränderte Attribute setzen. Werden neue Knoten generiert, so müssen die Attribute *id, file, llx, urx, lly* und *ury* gesetzt werden. Die Gruppenknoten erhalten mindestens die Attribute *id, file, llx, urx, lly* und *ury*. Beliebige weitere Attribute können sowohl bei den Einzel- als auch bei den Gruppenknoten gesetzt werden. Diese fließen dann in den weiteren Interpretationsvorgang ein.

4.4.4 Beschreibung der externen Operatoren

Um GEOAIDA über die korrekten Aufrufmodalitäten von Operatoren in Kenntnis zu setzen, werden alle verfügbaren Operatoren in XML-Dateien beschrieben, die beim Start von GEOAIDA eingelesen werden. Zur Operatorbeschreibung dient das Tag `<operator>`. Eigenschaften des Operators wie Name, auszuführendes Programm werden über Attribute beschrieben. Erlaubt der Operator eine Konfiguration über Kommandozeilenparameter, so werden diese Attribute über das Tag `<attribute>` beschrieben. Die mit `<attribute>` definierten Parameter können im Netzeditor von GEOAIDA als operator-spezifische Attribute gesetzt werden. Die `<attribute>`-Elemente werden in den zugehörigen `operator`-Tag eingebunden:

```
<operator name=... cmd=...>
  <attribute name=... type=... />
</operator>
```

Das Objekt `operator` hat die folgenden Attribute:

type: Der Typ eines Operators kann `topdown` oder `bottomup` sein.

class: Operatoren können in Klassen eingeteilt werden. Sollte die Zahl der Operatoren sehr groß werden, können diese bei Auswahl in Untermenüs gruppiert werden.

name: Über den Namen werden Operatoren eindeutig identifiziert und unter diesem dem Benutzer im Auswahlmenü angezeigt.

tip: Das `tip`-Attribut kann einen Hilfetext im HTML-Format enthalten, der dem Benutzer als Balloon-Help angezeigt wird.

cmd: Das Aufrufformat zur Ausführung wird in diesem Attribut festgelegt. Attribute aus dem semantischen oder dem Instanzen-Netz können in das Kommando übernommen werden, indem der Attributname in `@` eingeschlossen wird. Operator-spezifische Attribute werden auf die gleiche Weise eingebunden, müssen dann aber innerhalb der Operatorbeschreibung mit dem Tag `<attribute>` definiert werden.

runglobal: kann den Wert `true` oder `false` annehmen. Wird der `runglobal` auf den Wert `true` gesetzt, wird dem entsprechenden Operator das gesamte Bild bzw. die gesamten Bilder zur Bearbeitung übergeben. Das ist sinnvoll für Operatoren, die auf größeren Bildausschnitten besser arbeiten, da sie z.B. statistische Verteilungen im Bildmaterial ausnutzen.

Die in den Operator eingebetteten Objekte `attribute` besitzen die folgenden Eigenschaften:

name: Der Name eines Attributs wird für die Ersetzung innerhalb des Attribut- oder Operatorkommandos benötigt.

label: Unter dem zugewiesenen Label erscheint das Attribut für den Benutzer im GUI.

tip: Der Tipp kann dem Benutzer Hilfestellung zur Bedeutung des Attributs geben.

cmd: Ist die Eigenschaft `cmd` gesetzt, so werden bei Ausführung des Operators zunächst die Variablen in dem Attribut-`cmd`-String ersetzt, bevor dieser String dann als Ganzes in das Operatorkommando an die durch den Namen des Attributs gekennzeichnete Stelle eingesetzt wird. Bekommt das Attribut keinen Wert zugewiesen, bleibt auch die Eigenschaft `cmd` leer. Auf diese Weise ist es möglich, optionale Argumente im Operatorenkommando zu besetzen.

value: Über die Eigenschaft `value` kann dem Attribut ein Startwert zugeordnet werden, der übernommen wird, falls der Benutzer keinen anderen Wert setzt.

type: Der Typ eines Attributs kann `STRING`, `INT`, `DOUBLE`, `ENUM`, `BOOL` oder `IMAGE` sein. Abhängig vom Typ können weitere typspezifische Eigenschaften gesetzt werden.

min: Für Attribute vom Typ `INT` oder `DOUBLE`, kann der minimal zulässige Wert gesetzt werden.

max setzt den maximal zulässigen Wert für INT- oder DOUBLE-Attribute.

imagetype: Diese Eigenschaft kann nur für Attribute vom Typ IMAGE gesetzt werden. **imagetype** bestimmt, welche Art von Bildern diesem Attribut zugewiesen werden können. Derzeit sind die Bildtypen LASER, VIS, SAR, IR und VIDEO möglich. Im Prinzip ist jedoch jeder Bildtyp möglich, sofern in der Datenbank ebenfalls Bilder diesen Typs für die angewählte Region existieren.

options: Bei einem Attribut vom Typ ENUM können hier die möglichen Auswahloptionen angegeben werden. Es handelt sich dabei um eine durch Kommata getrennte Liste von Strings.

Neben operator-spezifischen Attributen, die über `<attribute>` innerhalb der `operator`-Objekte definiert werden, gibt es Attribute, die aus dem semantischen Netz und aus dem aktuellen Instanzennetz an den Operator über die Kommandozeile übergeben werden können. Dies sind einerseits Attribute, die allgemein den zu bearbeiteten Ausschnitt beschreiben, wie beispielsweise:

geoWest, geoNorth, geoEast, geoSouth : Geokoordinaten zu bearbeitenden Boundingbox,

minRes, maxRes : minimale und maximale Auflösung des zur Verfügung stehenden Bildmaterials,

input : Eingabedatei für die Knotenbeschreibung bei Bottom-Up-Operatoren.

output : Ausgabedatei für die Regionenbeschreibung bei Top-Down-Operatoren bzw. für die Knotenbeschreibung bei Bottom-Up-Operatoren.

Andererseits gibt es Attribute, die speziell das vom Operator angeforderte Bildmaterial beschreiben. Diesen Attributen wird jeweils der Name des IMAGE-Attributs des aufzurufenden Operators vorangestellt:

_file : Dateiname des Bildes

`_res_x`, `_res_y` : Auflösung des Bildes

`_size_x`, `_size_y` : Dimension des Bildes

Beispiel für eine Operatorbeschreibung Es existiere ein Operator zum Finden von Straßen *findStreet*. Das ausführbare Programm wird mittels `find_streets` aufgerufen. Über die Kommandozeile können die minimale und die maximale Straßenbreite angegeben werden. Wird die Option `--gis` verwendet, so benutzt der Operator vorhandene GIS-Daten zur Straßenextraktion aus der angegebenen Datenbank. Das Kommando wird folgendermaßen aufgerufen:

```
find_streets --min <wert> --max <wert> [--gis server] <vis-image> <output>
```

Die folgenden Zeilen beschreiben den Operator:

```
<operator name=findStreet class=bv type=topdown
  tip="Findet Straßen in visuellen Bildern<br>
    Optional kann eine GIS-Datenbank verwendet werden"
  cmd="find_streets --min @min@ --max @max@ @gis@ @vis@ @output@">
  <attribute name=min type=INT value=1.0 min=1.0 max=5.0 />
  <attribute name=max type=INT value=3.0 min=1.5 max=20.0 />
  <attribute name=gis type=STRING cmd="--gis @gis@" />
  <attribute name=vis type=IMAGE imagetype=VIS />
</operator>
```

Der Operator wird zwischen dem Starttag `<operator>` und dem Endtag `</operator>` definiert. Er erhält den Namen *findStreet* zugewiesen und wird in die Klasse *bv* (Bildverarbeitung) eingeordnet. Vom Typ her handelt es sich um einen Top-Down-Operator. Der im HTML-Format angegebene Tipp dient zur Information des Benutzers über die Bedeutung. Das Attribut `cmd` beschreibt den Aufruf des Programms. In `@...@` eingeschlossene Parameter werden vor Aufruf des Programms durch die aktuellen Knotenattribute im semantischen Netz ersetzt. In den `<attribute>`-Objekten werden die für diesen Operator spezifischen Parameter beschrieben. Diese erscheinen im Benutzerinterface des Netzeditors als top-down-operator-spezifische Attribute.

Wird der Operator nun während des Top-Down-Laufs aufgerufen, werden neben den operator-spezifischen Attributen `min`, `max`, `gis` die allgemeinen Attribute `geoWest`, `geoNorth`, `geoEast`, `geoSouth`, `minRes`, `maxRes` und `output` und die bildspezifischen Attribute `vis_file`, `vis_x_res`, `vis_y_res`, `vis_size_x` und `vis_size_y` gesetzt, die sich aus dem operator-spezifischen Attribut `vis` vom Typ `IMAGE` ergeben. Der Wert dieser Attribute wird an die entsprechenden Stellen der Kommandozeile eingefügt.

4.5 Ergebnisvisualisierung

Als Ergebnis der Analyse liefert GEOAIDA ein Instanzennetz, in dem alle bestätigten Objekte verzeichnet sind, und dazugehörige Labelbilder, in denen die örtliche Lage der erkannten Objekte beschrieben sind. Dieses Instanzennetz kann bereits bei kleinen Szenen sehr komplex werden, so dass sich die Frage stellt, wie diese Informationen dem Benutzer in geeigneter Weise dargestellt und wie die Informationen effizient verlustlos gespeichert werden können.

4.5.1 Datenhaltung

Da das finale Instanzennetz streng hierarchisch ist, d.h. untergeordnete Knoten immer in den übergeordneten räumlich enthalten sind, ist es möglich, die Lage sämtlicher Objekte in einem einzigen, zweidimensionalen Labelbild zu hinterlegen. Jedes Objekt muss neben seiner eigenen ID, unter der es im Labelbild abgelegt ist, auch sämtliche IDs der ihm untergeordneten Objekte kennen. Dies ist am effizientesten zu kodieren, indem jedes Objekt einen zusammenhängenden Block von IDs erhält, die es an die ihm untergeordneten Objekte vergeben kann.

Der Wurzelknoten, welcher im allgemeinen die Szene ist, erhält die ID 1. Die Vergabe der IDs erfolgt rekursiv durch Abarbeiten des Knotens I und dessen Subknoten I' . Dabei werden aufsteigende IDs vergeben. Im Knoten I wird die eigene ID als Start-ID und die ID, die nach Abarbeiten des letzten Subknoten generiert wurde, als End-ID abgespeichert. Alle Regionen mit IDs im Bereich

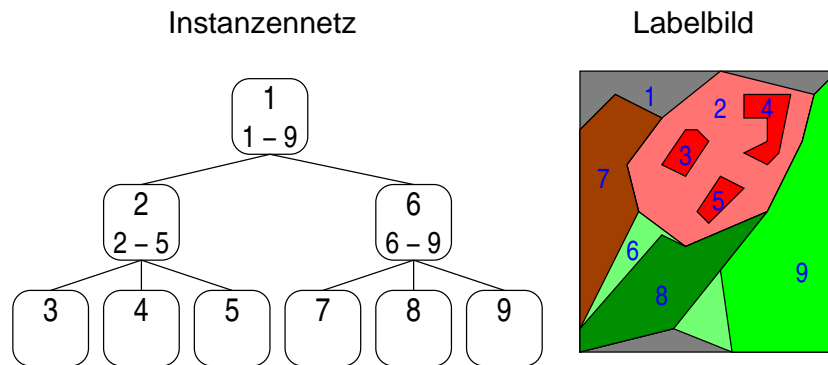


Abbildung 4.22: Vergabe der Label-IDs zum Erzeugen der Ergebniskarte: Im übergeordneten Knoten wird der ID-Bereich sämtlicher Subknoten vermerkt (links). Dies ist möglich, da die Subknoten jeweils vollständig in den übergeordneten Knoten enthalten sind (rechts).

der Start- und End-ID gehören zu der Region des Knotens I .

Abbildung 4.22 verdeutlicht das beschriebene Vorgehen. Zunächst erhält der Wurzelknoten die ID 1. Der linke Subknoten erhält die ID 2. Dessen Subknoten werden nun als nächstes abgearbeitet und erhalten die IDs 3 bis 5. 5 ist die zuletzt in diesem Zweig generierte ID, weshalb im Knoten 2 die ID 5 als End-ID eingetragen wird. Alle Knoten von 2 bis 5 gehören zur Region 2. Als nächstes wird nun der zweite Subknoten von Objekt 1 abgearbeitet und die ID 6 vergeben. Es folgen rekursiv dessen Subknoten, so dass schließlich dem Knoten 6 die IDs 6 bis 9 und der Szene 1 sämtliche Knoten von 1 bis 9 zugeordnet sind.

4.5.2 Interaktive Karte

Die im letzten Abschnitt beschriebene Datenhaltung ermöglicht nun die leichte Implementation einer interaktiven Karte. Die Karte besteht aus drei Komponenten, dem semantischen Netz, dem Instanzenetz und dem Labelbild, das die Lage sämtlicher Objekte beschreibt (s. Abbildung 4.23).

Die folgenden Funktionen sind implementiert:

Semantisches Netz

Labelbild

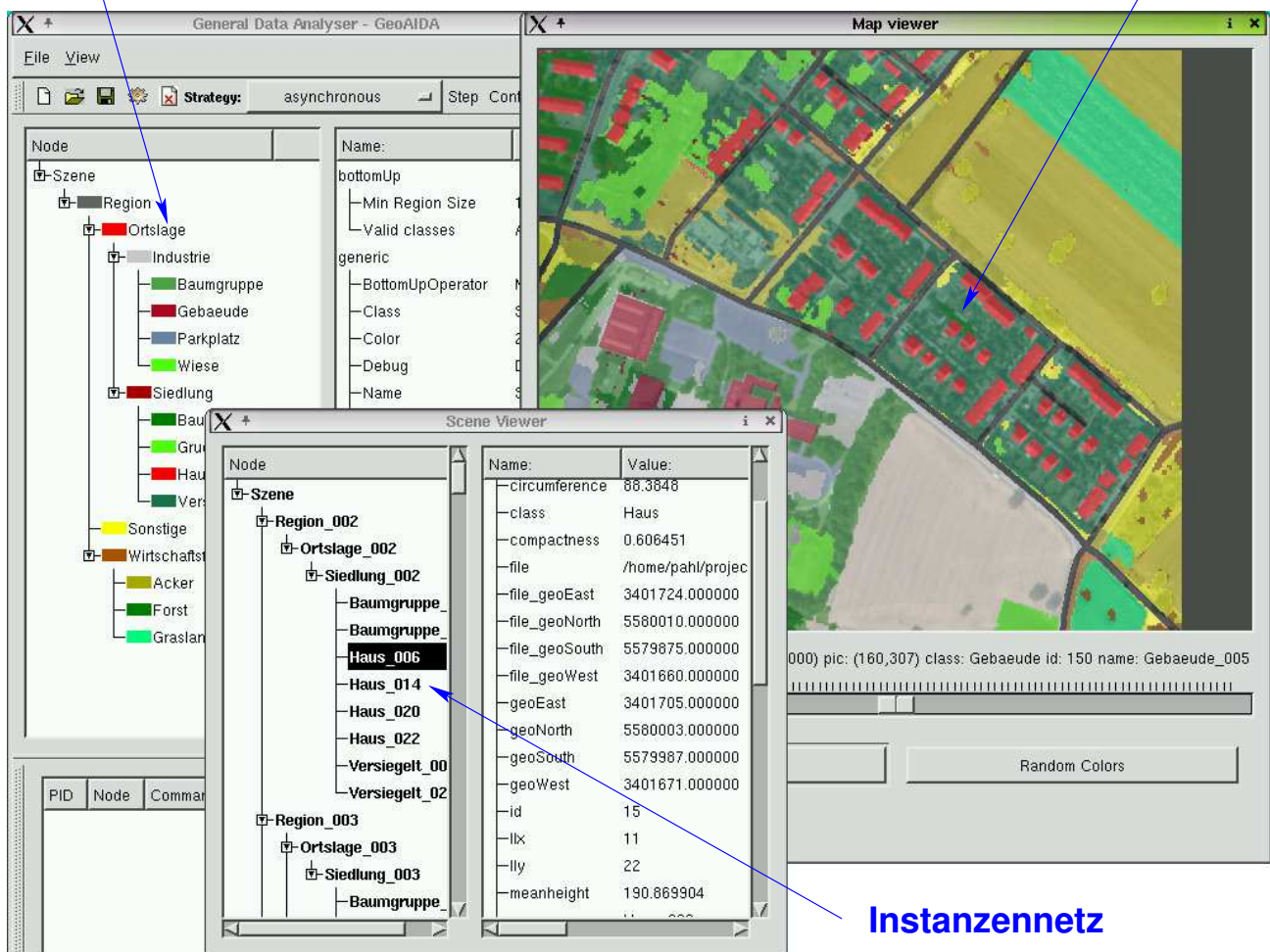


Abbildung 4.23: GEOAIDA-Benutzeroberfläche mit Darstellung der interaktiven Kartenfunktion.

- Wahl des Detaillierungsgrades im semantischen Netz mit automatischer Anpassung des Labelbildes und des Instanzennetzes,
- Wahl des Detaillierungsgrades für einzelne Regionen im Instanzennetz mit automatischer Anpassung des Labelbildes und
- Anzeige von Objekttypen und Objektattributen durch Anwählen eines Objektes im Labelbild.

In Abbildung 4.24 und 4.25 werden das Ein- bzw. Ausblenden von Teilbäumen im semantischen Netz und die Auswirkungen auf das Labelbild demonstriert.



Abbildung 4.24: *Interaktive Karte im Überblickszustand mit durchgeblendetem visuellen Bild (rechts) und dazugehöriges semantisches Netz (links).*

In Abbildung 4.24 sind nur Ortslagen (rot), Wirtschaftsflächen (braun) und sonstige Gebiete (gelb) dargestellt. In Abbildung 4.25 wurden sämtliche Details eingeblendet, indem die entsprechenden Objekte im semantischen Netz geöffnet wurden.

Abbildung 4.26 zeigt eine Karte in der lediglich eine Region im Instanzennetz geöffnet wurde. Hierzu wurden die Ortslagen zu Wirtschaftsflächen im semantischen Netz geschlossen und anschließend die detailliert dargestellte Ortslage im Instanzennetz wieder geöffnet.



Abbildung 4.25: *Interaktive Karte im Detailzustand mit durchgeblendetem visuellen Bild (rechts) und dazugehöriges semantisches Netz (links).*



Abbildung 4.26: *Überblickskarte mit durchgeblendetem visuellen Bild und Darstellung einer Region im Detail.*

4.6 Anforderungen an die externen Operatoren

Die eigentliche Bildverarbeitung ist in GEOAIDA in externe Programme ausgelagert, die als Operatoren bezeichnet werden. Die Anbindung der Operatoren geschieht unter vier Gesichtspunkten:

1. Es sollte leicht möglich sein, bestehende Bildverarbeitungsroutinen in GEOAIDA zu integrieren.
2. Viele Bildverarbeitungsroutinen sind nicht für die Verarbeitung von Fernerkundungsdaten entwickelt worden. Daher sollten fernerkundungsspezifische Eigenschaften, wie z.B. das Handling von Geokoordinaten keine Bedingung für die Integrationsfähigkeit einer Bildverarbeitungsroutine sein.
3. Manche Bildverarbeitungsroutinen arbeiten besser auf großen Bildausschnitten, da sie sich statistische Eigenschaften zu Nutze machen. Das Handling von Bildausschnitten sollte in diesem Fall von GEOAIDA übernommen werden.
4. Die Bildverarbeitungsroutinen sollten innerhalb von GEOAIDA kontextspezifisch konfigurierbar sein.

Wie die Anbindung und Konfiguration der Operatoren innerhalb von GEOAIDA technisch geschieht, wurde bereits im Abschnitt 4.4.4 beschrieben, wohingegen im folgenden die Funktionsweise der Operatoren näher erläutert werden soll.

Grundsätzlich wird in GEOAIDA zwischen zwei verschiedenen Arten von Operatoren unterschieden, an die unterschiedliche Anforderungen gestellt werden:

1. Top-Down-Operatoren sollen in einem festgelegten Weltausschnitt Objekte einer bestimmten Klasse liefern. Der Weltausschnitt wird dabei sowohl in Geokoordinaten als auch – falls vorhanden – in Form von Bildausschnitten bereitgestellt.

2. Bottom-Up-Operatoren erhalten die hypothetischen Objekte einer Region und müssen diese zu übergeordneten Objekten einer bestimmten Klasse gruppieren.

4.6.1 Bildinterpretationsoperatoren

Obwohl auch Anwendungen wie die Abfrage einer GIS-Datenbank denkbar sind, wird das häufigste Einsatzgebiet der Top-Down-Operatoren die Bildinterpretation sein. Sehr oft lässt sich die Bildinterpretation in die in Abbildung 4.27 dargestellten Schritte untergliedern.

Die Bildinterpretation besteht aus den Schritten *Segmentierung*, *Attributierung* und *Klassifizierung*. Die *Segmentierung* zerlegt das Bildmaterial in Regionen mit gleichen Eigenschaften bezüglich eines bestimmten Merkmals. Je nachdem, ob die Grenzen der Regionen oder der Zusammenhang von Pixeln innerhalb einer Region untersucht wird, unterscheidet man kontur- bzw. flächenbasierte Verfahren, von denen einige wichtige im Abschnitt 2.2 beschrieben wurden. Die *Attributierung* bestimmt für die Regionen Form- und Texturparameter. Aufgrund dieser Parameter versucht die *Klassifizierung* den Regionen eine Bedeutung zuzuweisen.

Häufig beherrschen vorhandene Bildverarbeitungsroutinen nur den ersten, den Segmentierungsschritt. Die weiteren Schritte lassen sich dann aber einfach durch Zusammenstellen mit den in GEOAIDA vorhandenen Modulen für Attributierung und Klassifizierung ergänzen.

Auch die Segmentierung besteht wiederum häufig aus Teilschritten. So wird auf die Eingangsdaten oft zunächst ein Rauschfilter angewendet, bevor die eigentliche Segmentierung startet. Die Fusion mehrerer Eingangsbilder erfordert einen Fusionsfilter usw. Jede dieser Funktionseinheiten lässt sich als ein Filter betrachten, in dem die Eingangsdaten eine irgendwie geartete, nicht notwendigerweise umkehrbare Transformation durchlaufen, um dann in einer Form zu erscheinen, die bezüglich der Gewinnung spezieller Informationen günstiger ist. Für den komplexen Segmentierungsschritt werden mehrere dieser Filter hintereinander geschaltet sein, um das gewünschte Ergebnis zu erzielen. Diese Sichtweise der

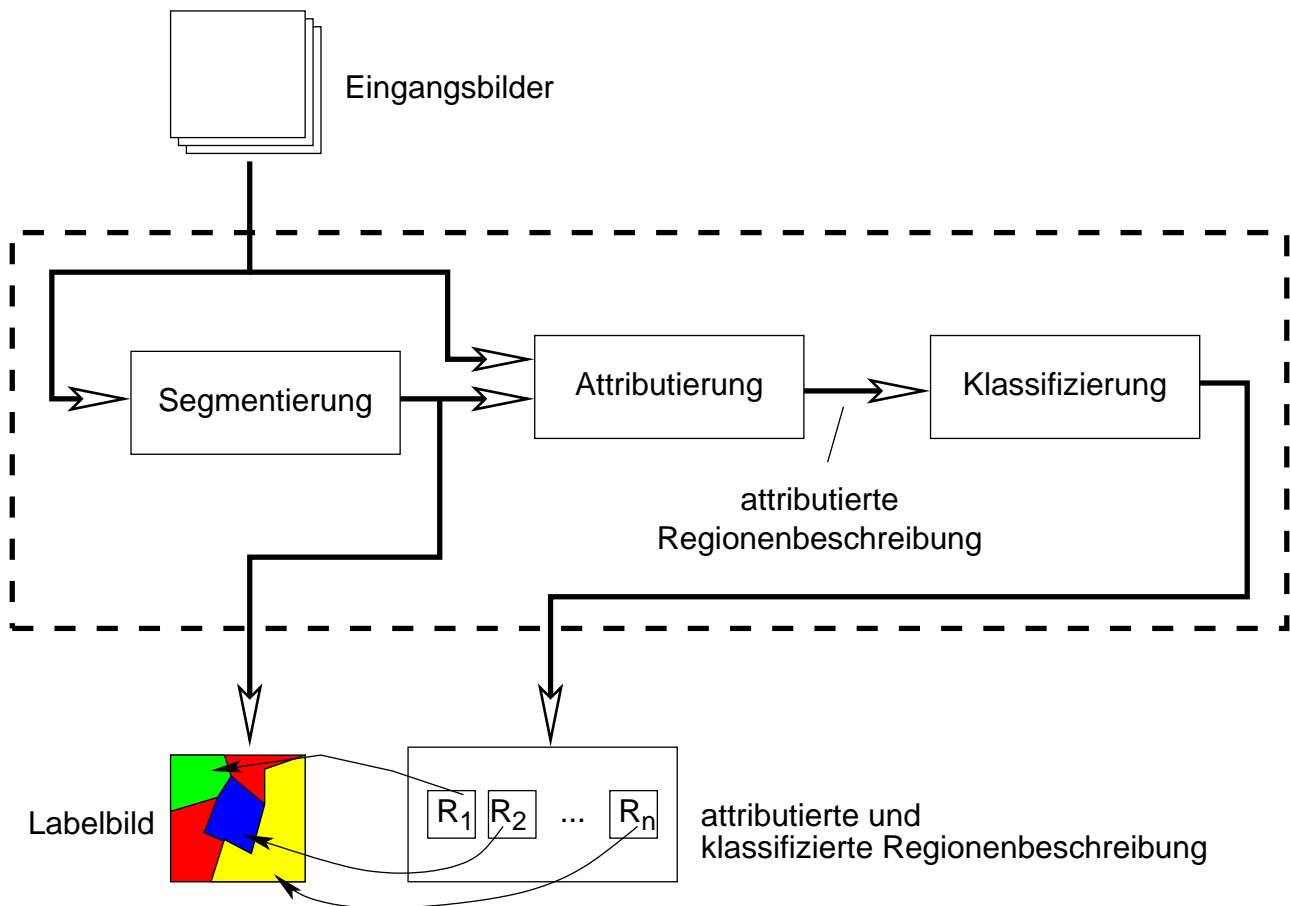


Abbildung 4.27: *Interner Aufbau eines Top-Down-Operators: Die Eingangsbilder werden zunächst segmentiert. Dann werden Attribute der einzelnen Segmente bestimmt. Anhand der bestimmten Attribute werden den Segmenten Klassen zugeordnet.*

Kopplung verschiedenartiger Filter macht sich das in Abschnitt 4.7 beschriebene Experimentalsystem zum Operatorentwurf zu Nutze.

Während die Attributierung eine lineare Abfolge von Schritten zur Bestimmung von Regionen-Parametern ist, die relativ unabhängig von der Aufgabenstellung ausgeführt werden kann, erfordert die Klassifizierung eine enge Anpassung an das gewünschte Interpretationsergebnis. Da hier eine generelle Strategie zur Problemlösung nicht angegeben werden kann, ist es sinnvoll, den Klassifizierungsschritt durch einen Interpretierer durchführen zu lassen, wie dieser auch für die Bewertung und Gruppierung in der Bottom-Up-Phase verwendet werden kann (s. Abschnitt 4.6.2).

Wie bereits zu Beginn dieses Abschnitts erwähnt, sind einige Bildverarbeitungsverfahren nicht in der Lage kleine Bildausschnitte zu bearbeiten, da sie auf statistische Eigenschaften angewiesen sind, die sich nur mit einer großen Stichprobe erreichen lassen. In diesem Fall wird zunächst das gesamte Bild interpretiert. Die Interpretationsergebnisse lassen sich dann über eine im Rahmen dieser Arbeit entstandenen Datenbank regionenbezogen abfragen.

Im Rahmen des Projekts GEOAIDA wurden einige Top-Down-Operatoren implementiert:

- ein auf der Varianz der Bilddaten arbeitender Operator, der zwischen künstlichen und natürlichen Objekten unterscheiden kann,
- eine Textursegmentierung basierend auf einem in [38] beschriebenen Verfahren,
- eine Straßenextraktion basierend auf einem in [24] beschriebenen Verfahren,
- eine Gebäudeextraktion, die im Rahmen dieser Arbeit entstanden ist (s. Abschnitt 5.1.1).

4.6.2 Bewertungs- und Gruppierungsoperatoren

Die Bottom-Up-Operatoren führen eine Bewertung der als Eingangsdaten bereitgestellten Liste von Knoten durch und gruppieren diese zu größeren Einheiten. Für die Bewertung der Knoten müssen bestimmte Eigenschaften der Knoten selbst oder im Verhältnis zueinander ermittelt werden. Bereits in [25] wurde hierfür eine erweiterbare Bibliothek entwickelt, die in der Lage ist, verschiedene Form- und Beziehungswerte zu errechnen. Auf Grund dieser errechneten Attribute werden schließlich die Bewertung und Gruppierung durchgeführt. Während das Errechnen der Attribute festen Regeln folgt, sind die anschließenden Auswerteregeln fallspezifisch und nicht parametrisierbar. Es wurde daher ein Interpreter entwickelt, der mit Hilfe einer Sprache in UPN⁷ Mengenverar-

⁷Umgekehrt Polnische Notation

beitung auf einem Stack durchführt. Dabei wird der in Abbildung 4.28 gezeigte Datenaufbau verwendet.

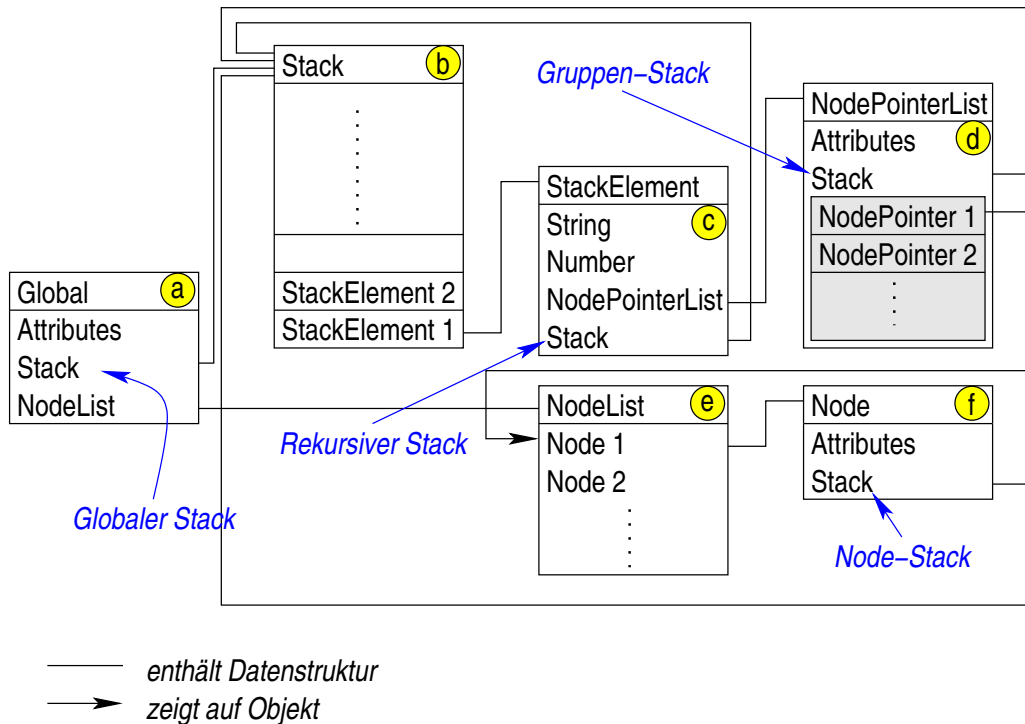


Abbildung 4.28: Datenstruktur des Interpreters für Objektmengenverarbeitung: In der globalen Datenstruktur (a) ist ein Stack (b) enthalten, der Elemente verschiedenen Typs aufnehmen kann, darunter eine Knotenliste (d), die selbst und deren Knoten wiederum ein Stack (b) beinhalten.

Der Interpreter selbst besitzt eine globale Datenstruktur (a), in der Attribute, ein Stack (b) und die eingelesene Knotenliste (e) enthalten sind. In der Knotenliste sind die Knoten (f) enthalten. Jeder Knoten besitzt wiederum Attribute und einen lokalen Stack (b). Der Stack (b) kann nun Elemente (c) des Typs *String*, *Number*, *NodePointerList* (d) und *Stack* (b) aufnehmen. Die Datenstruktur *NodePointerList* (d) dient als Gruppen-Container für Knoten. Die Knoten (f) selbst werden hier nur als Zeiger verwaltet, d.h. jeder Knoten ist physikalisch nur einmal im Speicher vorhanden, auch wenn er zwischenzeitlich mehreren Gruppen (*NodePointerList*) angehört. Zusätzlich kann die *NodePointerList* noch Attribute und einen Stack enthalten.

Der Interpreter erhält eine Befehlszeile, die er zunächst in einzelne Elemente (Token) zerlegt. Als nächstes wird der Typ des Token bestimmt. Handelt es sich um ein Datum (*Number*, *String*), so wird es auf dem globalen Stack abgelegt. Handelt es sich jedoch um eine Funktion, so wird diese ausgeführt.

Funktionen holen ihre Operanden vom globalen Stack, verarbeiten sie und schreiben ihr Ergebnis wieder auf den globalen Stack. Die spezielle Funktion `nodelist` platziert ein Stackelement des Typs *NodePointerList*, in dem sämtliche Knoten enthalten sind, auf den Stack. Funktionen, die mit dem Präfix `nl.` beginnen, erwarten auf dem globalen Stack ein Element vom Typ *NodePointerList* und führen ihre Funktionalität dann auf dem Stack dieses Elements aus. Gleiches gilt für Kommandos mit dem Präfix `node.`, jedoch werden die Funktionen auf sämtlichen Stacks der in der Gruppe befindlichen Knoten ausgeführt.

Neben den Grundoperationen, wie `+`, `-`, `*`, `/`, `<`, `>`, `=` gibt es Funktionen, die den Stack bearbeiten. `dup` erzeugt eine Kopie des obersten Stackelements. Die Varianten: `nl.dup` und `node.dup` arbeiten auf den entsprechenden lokalen Stacks. `drop` vernichtet das oberste Stackelement (Varianten wie bei `dup`).

Die `select`-Anweisung dient zur Auswahl von Knoten mit bestimmten Eigenschaften. Mit `set` und `get` (und ihren `nl.`- und `node.`-Varianten können Attribute gesetzt bzw. gelesen werden. Mit der `node.run`- und `nl.run`-Funktionen können Unterprogramme auf Knoten- bzw. Gruppenstacks ausgeführt werden. Spezielle Funktion errechnen Attribute wie Formparameter der Objekte oder Beziehungen zwischen den Objekten.

Das folgende Beispiel zeigt, wie die Segmente aus einem Farbbild in die Klassen *Unknown*, *Greenland* und *Shadow* eingeteilt werden. Es wird eine ungruppierte Knotenliste geschrieben, wie sie als Ergebnis eines Top-Down-Operators erwartet wird:

```
( 1) nodelist
( 2) "Unknown" node.push "class" node.set
( 3) "meancolor" node.get
( 4) "dup 80 > swap 280 < *" node.run
( 5) select
( 6) "Greenland" node.push "class" node.set
```

```
( 7) drop
( 8) nodelist
( 9) "meanbrightness" node.get 10 < select
(10) "Shadow" node.push "class" node.set
(11) drop
(12) nodelist
```

Zunächst wird die Knotenliste auf den Stack gebracht (Zeile 1). In Zeile 2 wird das Attribut *class* aller Knoten auf den Wert *Unknown* gesetzt. Zeile 3 legt den Wert des Attributs *meancolor*, welches den C-Kanal der Farbe im CSV-System enthält, auf den lokalen Stack sämtlicher Knoten. In Zeile 4 wird ein lokales Kommando auf den Knoten-Stacks ausgeführt, das ermittelt, ob sich die Farbe im Grünbereich befindet. Das Ergebnis dieser Kommandos führt in Zeile 5 zur Selektion der Grünflächen-Knoten. Für diese Knoten wird in Zeile 6 das Klassenattribut auf *Greenland* gesetzt. Zeile 7 nimmt die Grünland-Knoten vom Stack. In Zeile 8 wird eine Liste sämtlicher Knoten wieder auf den Stack gebracht. Diesmal wird die Helligkeit der Knoten untersucht (Zeile 9) und für dunkle Knoten die Klasse *Shadow* gesetzt (Zeile 10). Zeile 11 nimmt auch diese Liste vom Stack. Zeile 12 bewirkt, dass zum Schluss eine Liste sämtlicher Knoten auf dem Stack ist, die dann in die Ausgabedatei übertragen wird.

4.7 Experimentalsystem zum Operatorentwurf

Viele Operatoren lassen sich als eine Kombination mehrerer Filter betrachten. Zu Experimentalzwecken ist es sinnvoll, die Filter als eigenständige Programme zu implementieren und einzeln auszuführen. Auf diese Weise können Zwischenergebnisse kontrolliert werden und Teilschritte optimiert werden, ohne jedesmal die gesamte Verarbeitungskette durchzuführen. Sind die Filter erst einmal als eigenständige Programme ausgeführt, liegt es nahe, die Kombination der Filter in einer Benutzeroberfläche zu erstellen und dort auch die Ausführung zu kontrollieren. Die Idee ist nicht neu [2][1], jedoch erfordern die bekannten Systeme eine Einarbeitung in die mitgelieferten, speziellen Bildverarbeitungsbibliotheken. Routinen lassen sich schwer über Systemgrenzen hinweg kombinieren. In

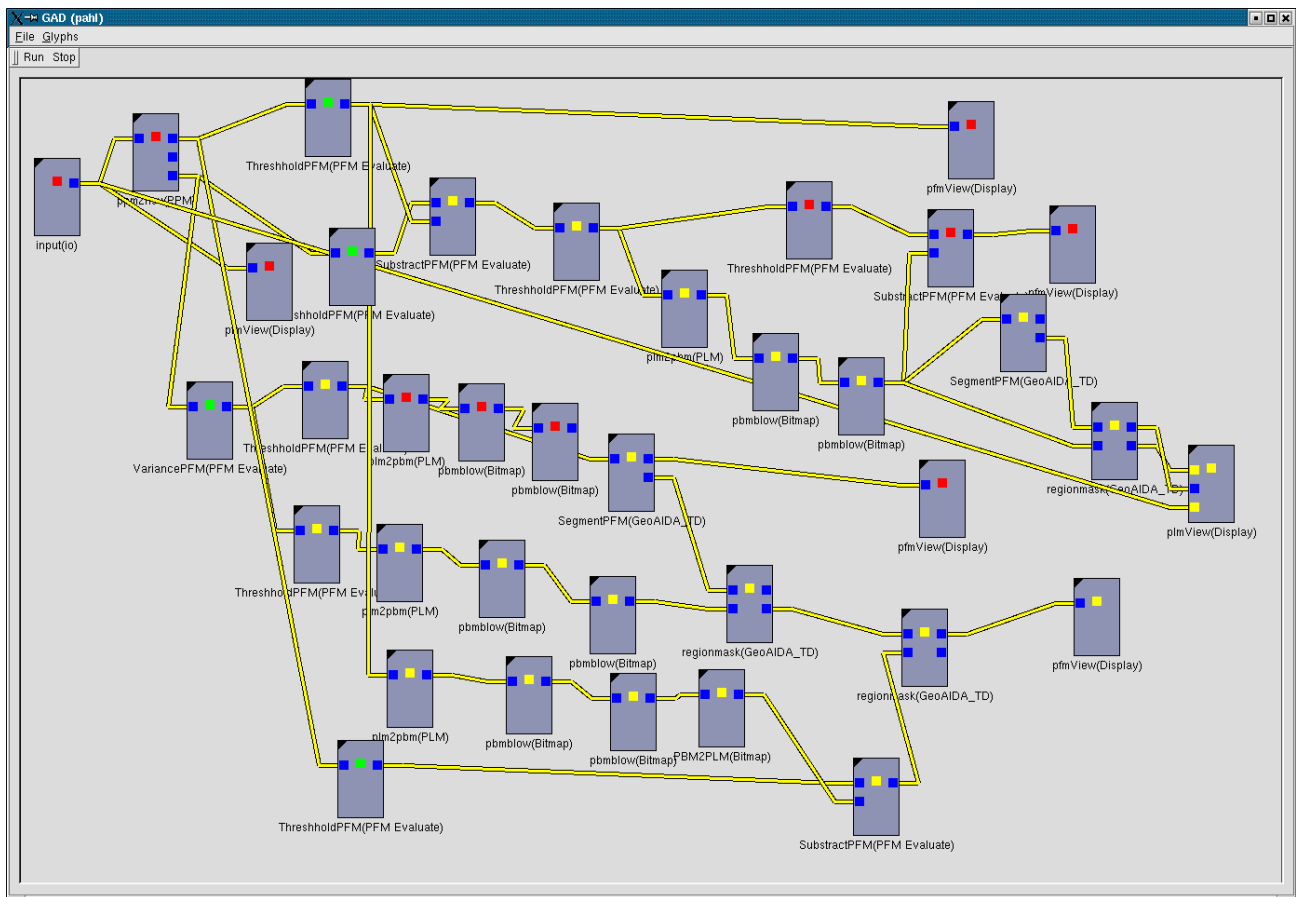


Abbildung 4.29: Im Experimentalsystem zum Operatorentwurf steht jeder der Boxen für ein ausführbares Programm, welches auf Knopfdruck aktiviert werden kann. Es werden dann die Abhängigkeiten zu anderen Programmen ermittelt und diese in der notwendigen Reihenfolge gestartet.

dem hier implementierten Experimentalsystem lassen sich beliebige Programme integrieren. Programme werden über eine XML-Beschreibung in die Oberfläche integriert. Die Beschreibung ist an die Operatorbeschreibung aus GEOAIDA angelehnt (s. Abschnitt 4.4.4). Parameter für die Programmausführung können innerhalb der Oberfläche verändert werden. Das System erkennt eigenständig, welche Teile des entworfenen Netzes bei Veränderung von Parametern oder Netzknoten neu bearbeitet werden müssen. Wird am Ende der Verarbeitungskette ein Filter eingesetzt, der eine zum GEOAIDA-System konforme Ausgabe erzeugt, kann die Verarbeitungskette leicht durch Umsetzung in ein Script in einen

Operator umgewandelt werden. In Abbildung 4.29 ist das Benutzerinterface des Experimentalsystems mit einem umfangreicheren Netz zu sehen.

5 Anwendungsbeispiele

GEOAIDA ist die Implementierung einer Architektur eines wissensbasierten Systems zur Steuerung der automatischen Interpretation von Fernerkundungsdaten. Als solches enthält es keinerlei Bildverarbeitungs- bzw. Bewertungsfunktionalität. Diese Funktionalität muss über externe Module bereitgestellt werden. Um die Leistungsfähigkeit des Systems zu zeigen, ist daher ein Einsatz von GEOAIDA in Anwendungsfeldern der Interpretation von Fernerkundungsdaten notwendig. In diesem Kapitel sollen zwei Anwendungsbeispiele vorgestellt werden, in denen GEOAIDA eingesetzt wird, eines aus dem Bereich der Landnutzungsanalyse aus Luftbildern in Kombination mit einem bestehenden GIS, das andere aus dem Bereich der Satellitenbilddauswertung. Zuvor soll jedoch auf einige im Rahmen des Projekts GEOAIDA entwickelte Operatoren zur Bildverarbeitung und Hypothesenbewertung eingegangen werden, die an verschiedenen Stellen der gewählten Anwendungsbeispiele eingesetzt werden.

5.1 Spezielle Operatoren

Im Rahmen des GEOAIDA-Projekts wurden Segmentierungsverfahren zur Detektion von Gebäuden in Laserscan-Höhendaten, für die Detektion von Straßenverläufen in Luft- und SAR-Bildern, zur Klassifizierung von Texturen in Form eines Lernverfahrens sowie eines auf der Texturvarianz beruhenden Verfahrens für den Einsatz als Top-Down-Operator für die Klassifizierung von Objekten aufbereitet bzw neu entwickelt. Desweiteren wurde ein allgemeiner Ansatz für die Hypothesenbewertung, der als Bottom-Up-Operator Verwendung findet, entwickelt. Da sie in den nachher beschriebenen Anwendungsbeispielen zum Einsatz kommen, sollen sie hier kurz beschrieben werden.

5.1.1 Gebäudedetektion in Laserscan-Daten

Flugzeuggestütztes Laserscanning ist inzwischen ein Standardverfahren, um eine dichtes DGM¹ zu erzeugen. Durch die gute Ortsauflösung und die hohe Messgenauigkeit eignen sich die Daten hervorragend zur Detektion von künstlichen Objekten. Etliche Arbeiten beschäftigen sich daher mit der Gebäudeextraktion aus Höhendaten [104][103][20][97]. In der Regel verwenden diese Verfahren parametrische Modelle, die an die Messdaten angepasst werden, oder benötigen die Grundrisse der Gebäude als Eingangsdaten.

Im Rahmen des GEOAIDA-Projekts wurde ein Verfahren entwickelt, das für die Gebäudedetektion lediglich die Höhendaten benötigt und nicht mit parametrischen Modellen arbeitet. Es handelt sich dabei um ein Region-Growing-Verfahren. Ausgehend von den Laserscan-Höhendaten werden zunächst Höhengsprünge detektiert. Zusammenhängende Regionen, bei denen sich benachbarte Höhenwerte um weniger als einen bestimmten Schwellwert unterscheiden, werden mit jeweils einem Label versehen. Zu kleine Regionen wie sie z.B. in Waldgebieten auftreten bleiben unklassifiziert, zu große Regionen werden dem Hintergrund zugeordnet.

Die Segmentierung erfolgt wie in Abbildung 5.1 gezeigt. Gestartet wird die Suche mit dem oberen linken Pixel im Bild. Die rechte Tabelle zeigt die Reihenfolge der Richtungswechsel an. Gestartet wird in der *Start*-Zeile mit der Richtung links. Ist das nicht möglich, wird nach unten gewechselt. Ist die Regionengrenze erreicht, so wird in die Reihe der Richtungstabelle gewechselt, deren linke Spalte die aktuelle Richtung enthält. Die dort verzeichneten gegenläufigen Richtungen werden getestet. Dieser rekursive Vorgang wird solange wiederholt, bis kein Richtungswechsel mehr möglich und die oberste Rekursionsebene erreicht ist. Es wird dann der nächste noch nicht einer Region zugehörige Pixel im Bild gesucht und dort entsprechend verfahren.

In Abbildung 5.2 wird die Anwendung des Verfahrens auf Laserscan-Daten der Karlsruher Innenstadt gezeigt. Die Höhendaten liegen im Raster von $0.5m$ mit einer Auflösung von $0.1m$ vor. Dabei wurde als Höhengschwellwert $1m$ verwendet. Mindestens 100 Höhenwerte und maximal 10000 Höhenwerte bilden eine Region.

¹Digitales Geländemodell

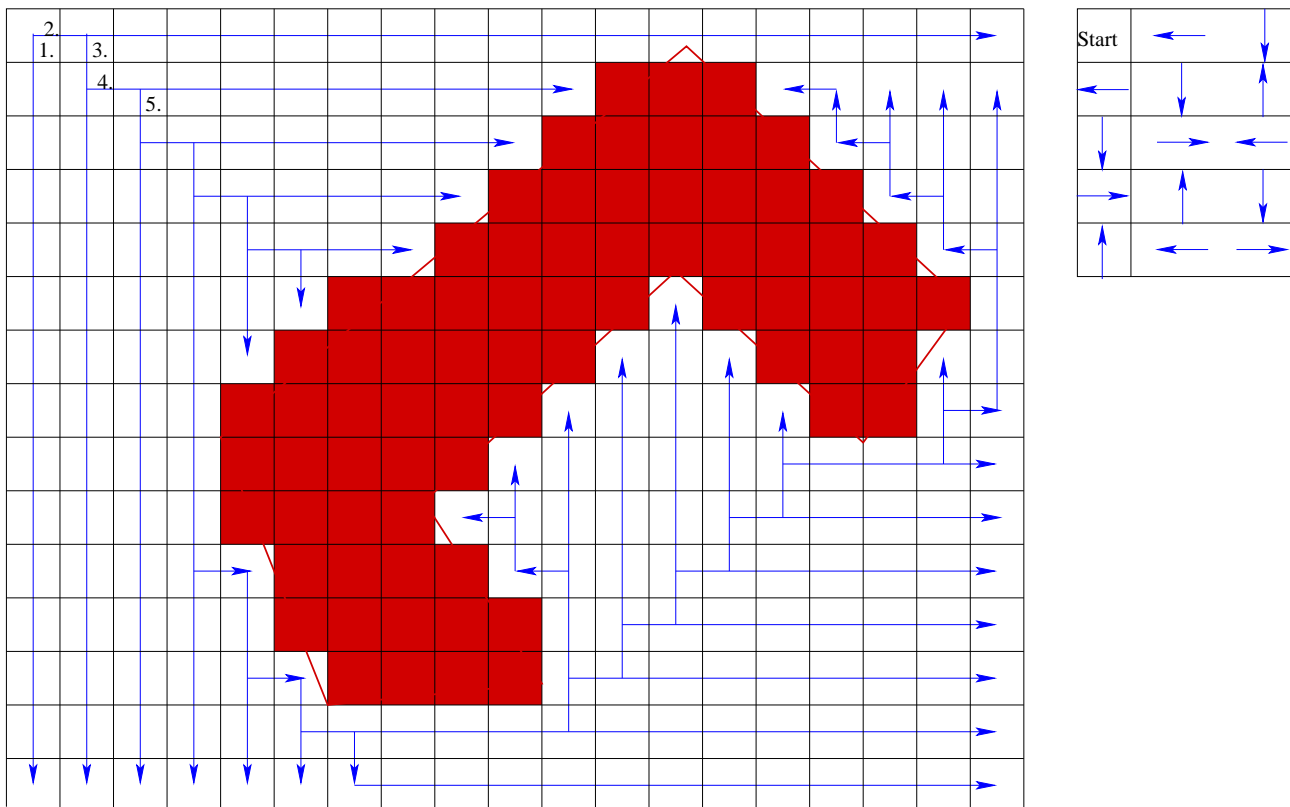


Abbildung 5.1: *Segmentierung von Gebäuden in Laserscan-Daten: Gestartet wird oben links im Bild mit den in der Tabelle gezeigten Startrichtungen. Die Arbeitsrichtung wird beibehalten, solange die Pixel auf dem Weg zu Region gehören. Danach wird entsprechend der Tabelle die nächste Arbeitsrichtung ausgewählt.*

Wie die Abbildung zeigt, werden durch diese Parameterwahl im wesentlichen Gebäude und geschlossenen Innenhöfen erkannt. Waldgebiete und Grünflächen fallen in der Regel heraus, da sie keine genügend großen, zusammenhängenden Regionen darstellen.

Ein geschlossener Innenhof zeichnet sich dadurch aus, dass er von höherliegenden Plateaus umgeben ist. Um die Klasse *Hof* gegenüber der Klasse *Gebäude* abzugrenzen, sind daher zunächst die Nachbarregionen einer Region zu bestimmen. Liegen die Nachbarregionen sämtlich höher als die betrachtete Region, so wird die betreffende Region als *Hof* klassifiziert. Abbildung 5.3 zeigt das Ergebnis dieser Klassifikation am Beispiel der Innenstadt von Karlsruhe. Die Klasse *Gebäude* ist in rot, die Klasse *Hof* in grün dargestellt.

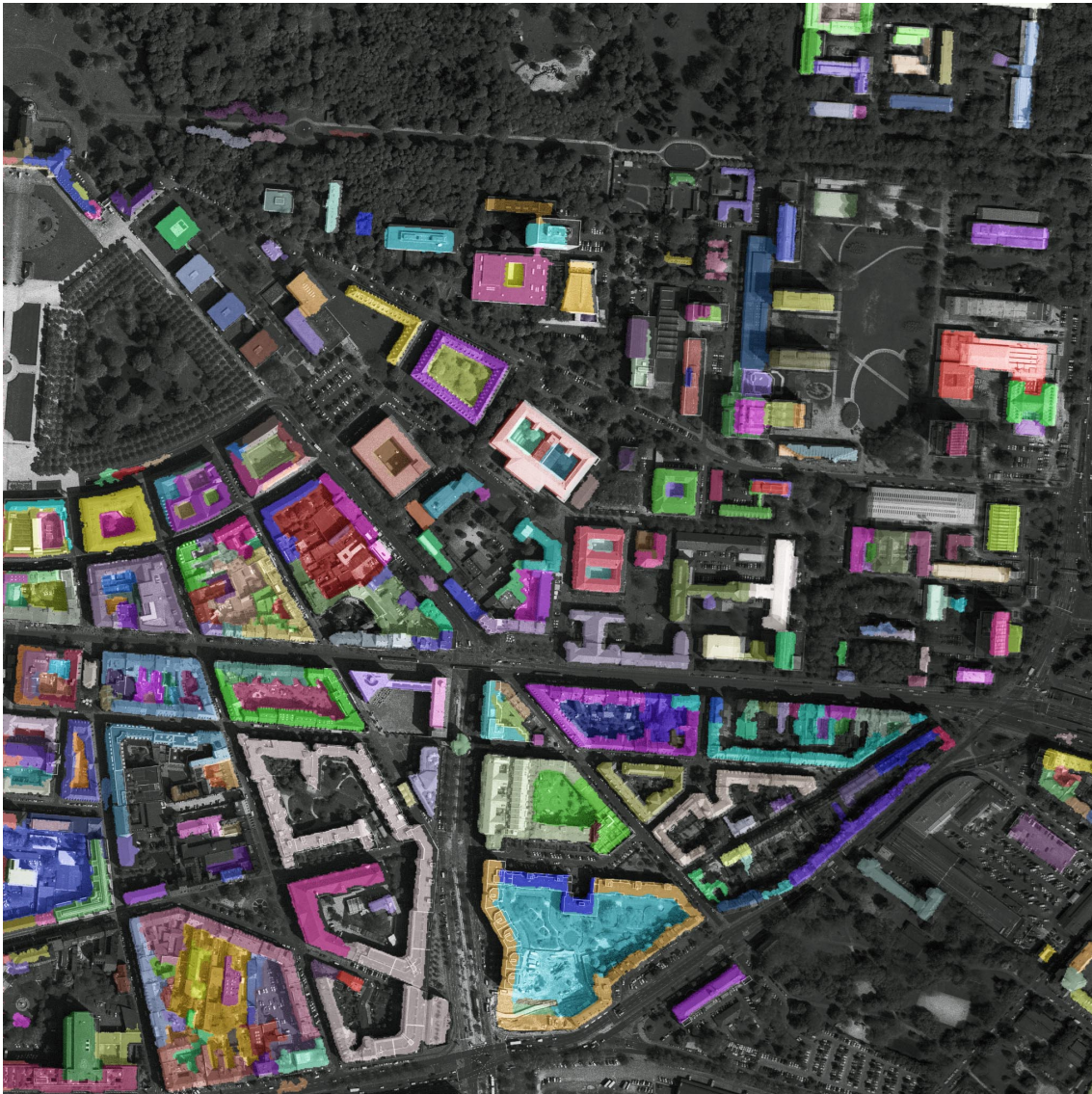


Abbildung 5.2: *Ergebnis der Segmentierung von Höhenplateaus in Laserscandaten*

5.1.2 Texturbasierte Klassifizierung

Bilder natürlicher Oberflächen können effizient durch Texturmodelle beschrieben werden, die auf zweidimensionalen stochastischen Prozessen basieren. Eine mögliche Implementierung eines stochastischen Modells für die überwachte Texturklassifizierung ist in [38] beschrieben. Überwacht bedeutet in diesem Zusammenhang, dass eine Texturprobe, welche die im Bildmaterial zu erwartenden Klassen beinhaltet, hinsichtlich ihrer stochastischen Merkmale erster und zwei-

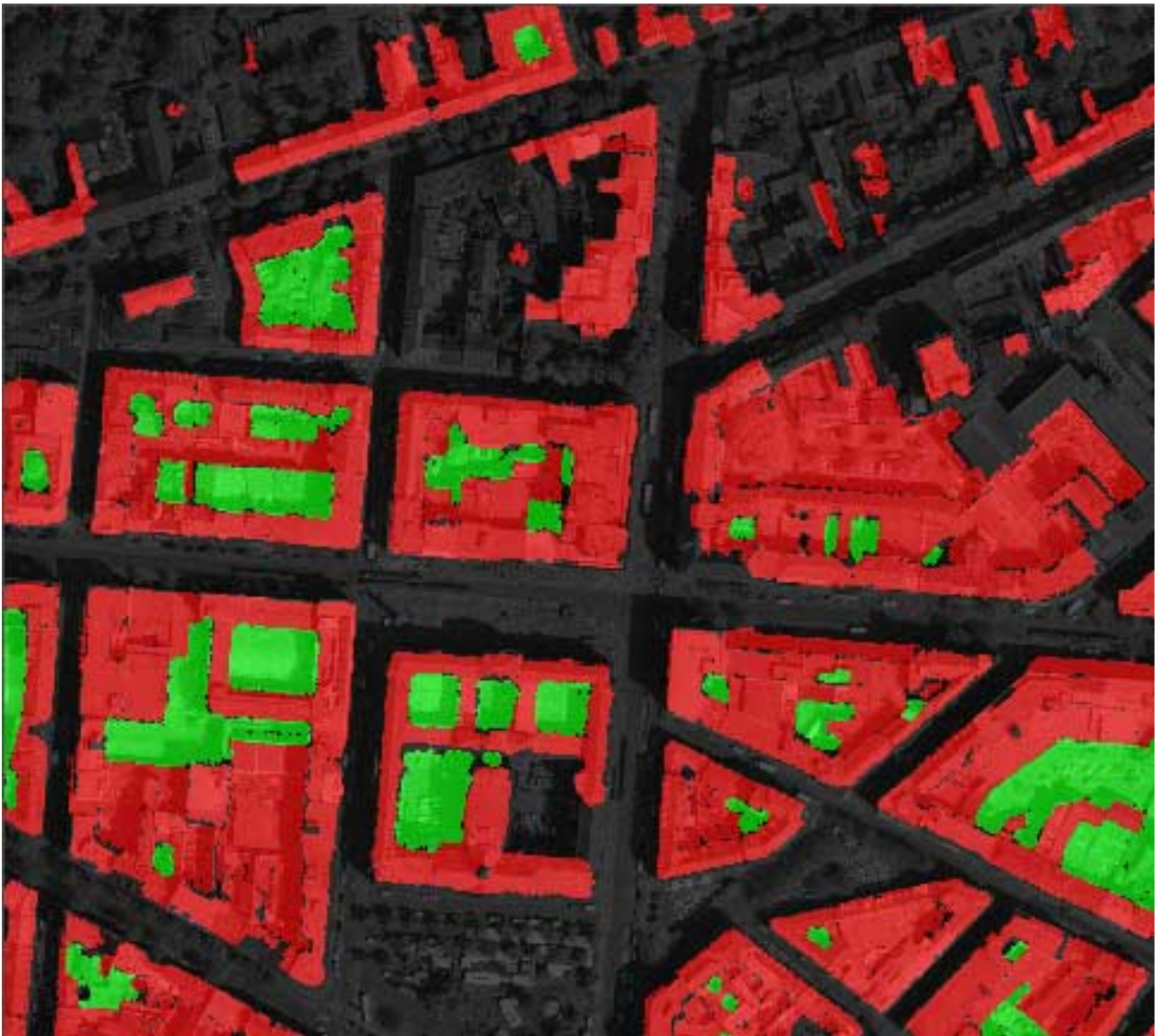


Abbildung 5.3: *Segmentierung von Laserscandaten mit anschließender Klassifikation in Gebäude (rot) und Hof (grün).*

ter Ordnung untersucht wird. Das Modell erster Ordnung beschreibt dabei die Wahrscheinlichkeitsdichte des Texturprozesses. Die Dichte wird aus dem Luminanzhistogramm aller Elemente jeder Trainingsklasse ermittelt.

In der Regel ist das Auftreten eines bestimmten Luminanzwertes statistisch aber von den Werten seiner Nachbarn abhängig. Das Texturmodell zweiter Ordnung berücksichtigt daher die paarweise Abhängigkeit von benachbarten Pixeln, sogenannten Cliques, und beschreibt diese lokalen Abhängigkeiten durch Gibbs-

Zufallsfelder. Nur statistisch signifikante Cliques-Typen werden für die Definition des Nachbarschaftssystems κ ausgewählt. Das Texturmodell besteht aus vier Wahrscheinlichkeiten:

- Luminanzwahrscheinlichkeit P_1 für jede Klasse (1. Ordnung),
- Wahrscheinlichkeit P_2 für jede Clique, dass sie derselben Klasse angehören,
- Wahrscheinlichkeit P_3 für die Luminanzdifferenz jeder Clique innerhalb einer Region und
- Wahrscheinlichkeit P_4 für die Luminanzdifferenz jeder Clique an der Regionengrenze.

Bei mehrbändigen Bildern wird zudem die Abhängigkeit der Bänder untereinander untersucht. Die Wahrscheinlichkeiten P_i bilden das Potential V_κ einer Clique κ . Die logarithmische Wahrscheinlichkeit wird als Potentialfunktion genutzt, so dass die Multiplikation für Verbundwahrscheinlichkeiten als Summe der gewichteten Potentiale betrachtet werden kann:

$$V_\kappa(x|k) = - \sum_i w_i \ln(P_i(x|k))$$

Gibbs-Zufallsfelder beschreiben die Verbundwahrscheinlichkeit einer Segmentierung eines Nachbarschaftssystems κ :

$$P(k|x) = \frac{1}{Z} e^{-\sum_x V_x(x|k)}, Z = \text{norm.}$$

Aufgabe der Segmentierung ist es die Verbundwahrscheinlichkeit für das ganze Bild zu maximieren. Bedingt durch Nachbarschaftsrelationen ist dies ein iterativer Prozess.

5.1.3 Klassifizierung über Varianz

Die Varianz eines Bildes kann als Maß für dessen Stukturiertheit benutzt werden. Mit sinkender Auflösung des Bildmaterials ist eine hohe Varianz ein Indikator für größere Strukturen. Empirisch wurde ermittelt, dass künstliche Objekte wie beispielsweise Gebäude in einer Bildauflösung von mehr als einem Pixel/Meter besonders hohe Varianzen verursachen.

Daher wurde im Rahmen dieses Projekts ein Segmentierungsoperator auf Basis der Varianz entwickelt, der es erlaubt Suchbereiche für künstliche Objekte zu bestimmen. Zunächst wird dazu die Varianz auf einem Fenster vorgebarer Größe für das gesamte Bild bestimmt. Über eine Analyse des Histogramms der Varianz (s. Abschnitt 2.2.1) wird eine initiale Segmentierung durchgeführt.

Da die Varianz der Varianz in bebauten Gebieten ebenfalls sehr hoch ist, ist die so entstandene Segmentierung sehr „löchrig“. Dieser „Löcher“ werden geschlossen, indem kleine zusammenhängende Gebiete ihrer Umgebung zugerechnet werden.

Um einen möglichst zusammenhängenden Suchraum zu erhalten, werden als nächstes dicht beieinander liegende Regionen verbunden. Sind zwei Regionen mehrfach miteinander verbunden, so entstehen erneut „Löcher“, die wiederum geschlossen werden. Im letzten Schritt werden feine Strukturen entfernt.

Dieses einfache Verfahren ist besonders interessant für die schnelle Suchraumeingrenzung für die Klassifizierung urbaner Gebiete. Speziellere und aufwändigere Operatoren können dann beispielsweise durch die Detektion von Gebäuden die vorgeschlagene Interpretation verifizieren bzw. falsifizieren.

5.1.4 Detektion von Straßen

Für die Detektion von Verkehrswegen wurde in [24] ein Verfahren entwickelt und im Rahmen von [25] als Top-Down-Operator in GEOAIDA integriert. Dieses nutzt das typische Luminanzprofil in Querrichtung und die Homogenität und Kontinuität in Längsrichtung aus, mit der sich die Verkehrswege im Bildmaterial ausprägen. Der Algorithmus gliedert sich in drei Teile. Im ersten Schritt wird

ein Gradienten- und Richtungsbild berechnet. Mit Hilfe dieser Information werden Gebiete identifiziert, die einem vorgegebenen parametrischen Nutzerprofil entsprechen. Beispielsweise werden Kandidaten für Straßenabschnitte unterhalb einer vorgegebenen Länge nicht akzeptiert. Weitere typische Parameter sind ein Straßenbreitenintervall, ein Helligkeitsintervall sowie ein Gradientenprofil gemäß Abbildung 5.4.

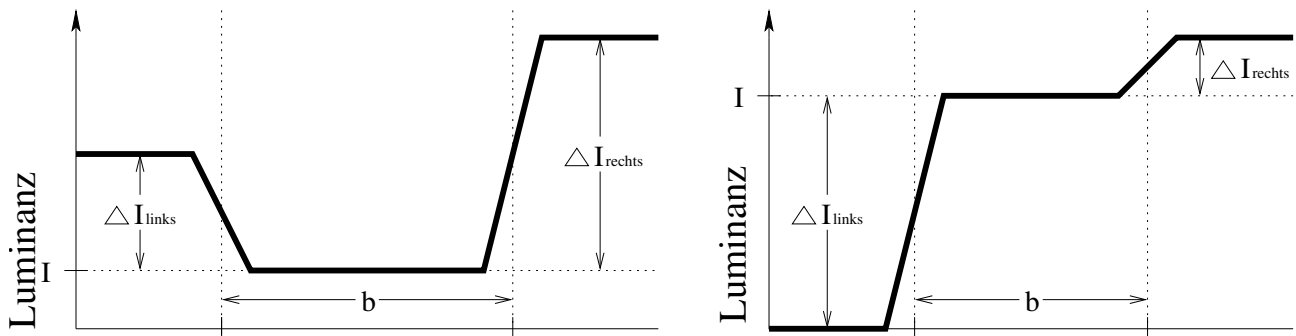


Abbildung 5.4: Gradient der vertikalen Straßenprofile.

Im zweiten Teil des Verfahrens werden die im ersten Schritt ermittelten Straßenkandidaten mittels eines A*-Suchalgorithmus erweitert. Mittels dieser Suche wird versucht, den besten Weg durch das Gradientenbild in Bezug auf das vorgegebene parametrische Straßenmodell zu finden. Die Suche wird iterativ solange fortgesetzt, bis eine vorgegebene Suchtiefe s erreicht ist.

Abbildung 5.5 verdeutlicht den Suchvorgang. In dem Beispiel stehen drei Alternativen für eine Erweiterung des Straßenabschnitts zur Verfügung. Eine Straßenerweiterung endet, wenn ein bestimmter Kostenwert überschritten ist (s. Gleichungen 5.1-5.3). Die Gesamtkosten c_{total} setzen sich aus der Summe der bereits akkumulierten Suchkosten c_{accum} und einer gewichteten Schätzung der Restkosten c_{remain} zusammen. w bezeichnet dabei die zurückgelegte Suchtiefe, s die maximal zulässige Suchtiefe und λ einen Gewichtungsfaktor. $c_{element}$ steht für die Kosten eines Wegelementes.

$$c_{total} = c_{accum} + c_{remain} \quad (5.1)$$

$$c_{accum} = \sum_w c_{element} \quad (5.2)$$

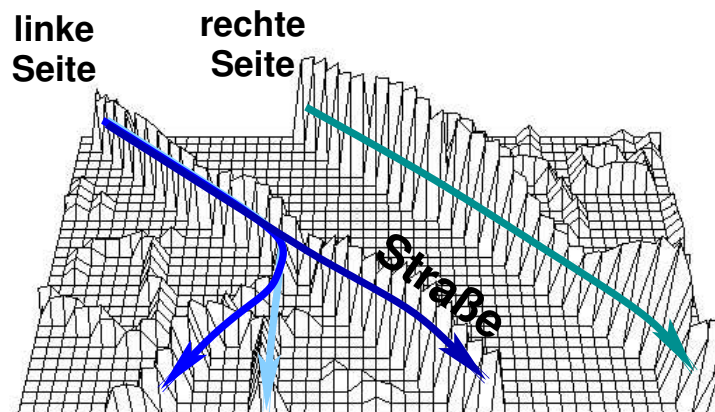


Abbildung 5.5: Suche des optimalen Straßenverlaufs.

$$c_{remain} = \lambda \frac{s - w}{w} c_{accum} \quad (5.3)$$

Im letzten Schritt des Algorithmus werden die gefundenen Liniensegmente vektorisiert. Damit verbunden ist eine einstellbare Glättung der Straßenverläufe.

Durch Parameterwahl ist das Verfahren sowohl in SAR- als auch in RMK-Bildern einsetzbar.

5.1.5 Ein generischer Bewertungsoperator

Generell sind beliebige Bewertungsoperatoren in GEOAIDA einbindbar. Die Praxis hat jedoch gezeigt, dass sich sehr viele Bewertungs- und Gruppierungsaufgaben mit einem kleinen Satz an Funktionen in unterschiedlicher Kombination lösen lassen. Dies führte in [25] zur Entwicklung eines generischen Bewertungsoperators.

Dabei war das Problem zu lösen, dass der Satz an benötigten Funktionen zwar relativ fest ist, die benötigte Kombination jedoch anwendungsabhängig. Dies führte zu der Lösung, die Funktionen zur Bestimmung von Objektattributen – wie Form und Farbe – und zur Gruppierung aufgrund von Objektrelationen – wie „nahe beieinander“ – in der compilerbasierten Sprache C++ effizient zu im-

plementieren, die Kombination der Funktionen über den Editor für semantische Netze innerhalb von GEOAIDA aber dem in 4.6.2 beschriebenen Interpreter zu übertragen. Da die Funktionsbibliothek und der Interpreter getrennte Einheiten sind, die über ein definiertes Interface miteinander verbunden sind, sind Funktionserweiterungen einfach in den Interpreter einzubinden.

5.2 Beispiele zur luftbildgestützten Landnutzungsanalyse

In [25] wird anhand mehrerer Beispiele beschrieben, wie die in dieser Arbeit entwickelte Architektur zur Interpretation von Fernerkundungsdaten die Landnutzungsanalyse auf Basis von ATKIS-Objektklassen effizienter gestalten kann. Dabei wird auf die Problemfelder Analyseablauf, Bewertung von Hypothesen mittels eines generischen Bewertungsoperators, Behandlung von Alternativen, datengetriebene Aufspaltung von Regionen, topologische Beziehungen von Objekten einer Ebene und Spezialisierung von Objekten eingegangen.

Als Vorbereitung für die Landnutzungsanalyse wird zunächst ein an der ATKIS-Hierarchie angelehntes semantisches Netz entworfen (s. Abbildung 5.6). Die Objektklassen *Ortslage*, *Wirtschaftsfläche*, *Siedlung*, *Industrie*, *Acker*, *Grasland*, *Wald* entstammen direkt dem ATKIS-Katalog.

Oftmals werden unterschiedliche Landnutzungen durch Straßen begrenzt. Daher wurde der Knoten *Region* eingefügt. Die initialen Regionen lassen sich über das Straßennetz erzeugen, welches entweder einem GIS entnommen oder mit dem in Abschnitt 5.1.4 beschriebenen Straßendetektionsalgorithmus gesucht wird. Suchbereiche für die Objektklassen *Ortslage* und *Wirtschaftsfläche* lassen sich über eine Varianzanalyse gemäß Abschnitt 5.1.3 bestimmen, da die Varianz in niedrigeren Bildauflösungen in bewohnten Gebieten bedingt durch künstliche Bauten höher ist als in unbewohnten. Eine weitere Möglichkeit, Hinweise auf eine Ortslage zu erhalten, ist die Regionen auf möglicherweise vorhandene Gebäude zu untersuchen (s. Abschnitt 5.1.1).

Die Unterscheidung zwischen den Klassen *Acker*, *Grasland* und *Wald* geschieht

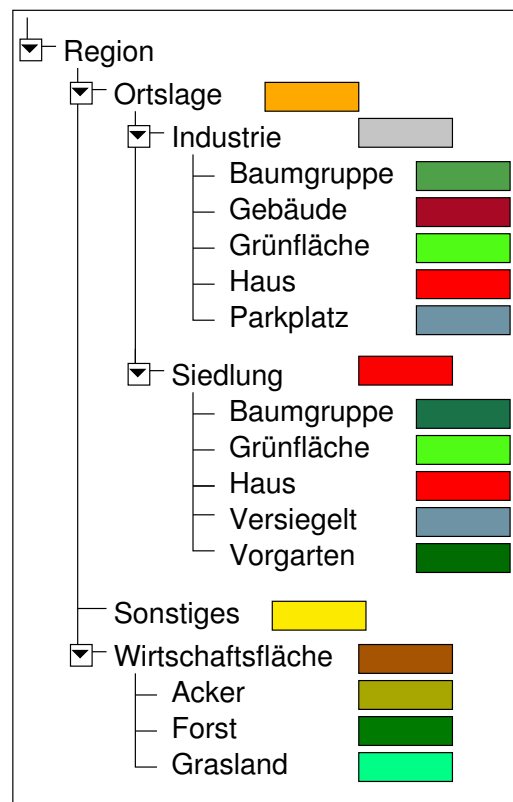


Abbildung 5.6: ATKIS-Objekte umgesetzt in einem semantischen Netz: Die Klasse *Region* dient als Hilfsklasse zur initialen Aufteilung der Szene. Die Objekte unterhalb der ATKIS-Klassen *Industrie* und *Siedlung* werden für eine strukturelle Analyse benötigt, da die Klassen nicht direkt erkannt werden können.

mit Hilfe des in Abschnitt 5.1.2 beschriebenen Verfahrens zur texturbasierten Klassifikation. Dieses Verfahren wird ebenfalls zur Detektion der Klassen *Garten* und *Parkplatz* eingesetzt. Die Klassen *Gebäude* und *Haus*, die sich durch Größe und Form unterscheiden, werden mit Hilfe des in Abschnitt 5.1.1 beschriebenen Verfahrens klassifiziert. Die Erkennung der Klassen *Siedlung* und *Industrie* geschieht rein strukturell über ihre Bestandteile.

5.2.1 Hierarchischer Analyseablauf und Bewertung von Alternativen

Von der in Abbildung 5.7 dargestellten Szene aus Koblenz liegen ein Luftbild, die Höheninformation sowie Informationen über das Straßennetz, entnommen aus dem ATKIS vor.



Abbildung 5.7: *Eingangsdaten der Szene Koblenz: VIS und Laserscan.*

Abbildung 5.8 zeigt einige Teilergebnisse der Interpretation. Zunächst wird das Bildmaterial mittels des Straßennetzes in Teilregionen geteilt. Eine dieser Teilregionen ist in der Abbildung 5.8a dargestellt. Mittels einer Varianzanalyse (s. 5.1.3) wird der Suchraum für die gezeigte Region auf die Teilhierarchie *Ortslage* eingeschränkt, so dass eine Entscheidung zwischen *Siedlung* und *Industrie* zu treffen ist. Für die Teilregion wird daher die Detektion von Gebäuden angestoßen. Das Ergebnis ist in Abbildung 5.8b mit durchgeblendeten Luftbild sowie in Abbildung 5.8c als Karte zu sehen. Die Bewertung, ob es sich bei der gezeigten Teilregion um ein Siedlungsgebiet handelt, wird nun mittels des generischen

Bewertungsoperators aus 5.1.5 durchgeführt. Dieser bewertet zunächst Größe und Form der gefundenen Häuser. Anschließend wird der Bebauungsanteil an der Gesamtfläche ermittelt. Dies führt schließlich zur Entscheidung, die Region gemäß Abbildung 5.8d als Siedlung zu instanzieren. Die Interpretation der Region als Industriegebiet scheiterte an der Detektion von Gebäuden, welche die für Industrieanlagen nötigen Ausmaße aufweisen.



Abbildung 5.8: *Teilinterpretation der Szene Koblenz: a) initiale Regionenaufteilung anhand des Straßennetzes, b) in das Luftbild eingebblendete erkannte Siedlungsobjekte, c) Siedlungsobjekte in Kartenform, d) finales Ergebnis der Landnutzungsanalyse.*

In Abbildung 5.9 sind sowohl Hinweise auf Siedlungsgebiete (links) als auch auf Industrieanlagen (rechts) zu finden. Die Legende zu diesen Karten findet sich in Abbildung 5.6. Zwei Lösungsstrategien bieten sich in diesem Fall an: Es kann eine Entscheidung zu Gunsten einer der beiden Alternativen getroffen werden oder die Region wird geteilt. Spielt die Anzahl der Häuser im Verhältnis zur flächenhaften Ausdehnung der Industrieanlagen nur eine geringe Rolle, so ist davon auszugehen, dass es sich trotzdem um ein Industriegebiet handelt, denn auch in Industriegebieten treten vereinzelt kleinere Gebäude auf. Diese Lösung ist für die gezeigte Situation zu wählen. Die zweite Lösungsmöglichkeit wäre die Teilung der Region in Siedlung und Industrie. Beide Lösungsansätze lassen sich mittels des in 5.1.5 beschriebenen generischen Bewertungsoperators realisieren.

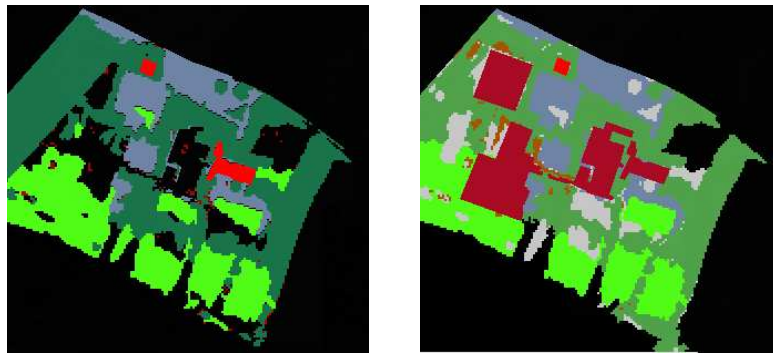


Abbildung 5.9: *Alternative Interpretation einer Teilregion als Siedlungs- (links) bzw. Industriegebiet (rechts). Die Legende ist in Abbildung 5.6.*

5.2.2 Vergleich eines Interpretationsergebnisses mit ATKIS



Abbildung 5.10: *Eingangsdaten der Szene Ravensburg: VIS, IR und Laserscan.*

Neueste Sensoren erlauben die gleichzeitige Erfassung von Laserscan-Höheninformationen, Infrarot- und Farbbildern auf einer Plattform, indem sie einen Laserscanner mit einer Mehrkanal-Zeilenkamera synchronisieren. Dies bietet zwei Vorteile. Zum einen ist eine Georeferenzierung der Daten untereinander überflüssig, wie dies bei zeitversetzter oder getrennter Erfassung in mehreren Sensoren bisher erforderlich war. Zum anderen wird durch die gleichzeitige Erfassung der Bild- und Höhendaten eine Entzerrung der Bilddaten erheblich erleichtert, da der eingesetzte Sensor im gesamten Scanbereich einen sehr steilen Einfallswinkel besitzt und zudem zu jedem Bildpunkt die Höhe bekannt ist. In Abbildung 5.10 ist ein Ausschnitt eines multisensoriellen Datensatzes aus einer Szene in Ravensburg zu sehen. Er besteht aus einem visuellem und einem Infrarotbild mit einer Auflösung von 1Pixel/25cm und einem Laserscan-Höhenmodell mit einer Auflösung von 1Pixel/m.

Durch die Verfügbarkeit des IR-Kanals wird die Landnutzungsanalyse zuverlässiger, da sich in Kombination mit dem visuellen Bild Vegetationsbereiche gut von versiegelter Fläche unterscheiden lassen. Zudem ermöglicht die Hinzunahme von Laserscan-Höhendaten das Ermitteln der Vegetationshöhe, da Vegetation für Licht teildurchlässig ist und somit Teile des Laserstrahls in unterschiedlichen Höhen der Vegetation reflektiert werden. Standardmäßig wird beim Laserscanning der erste und letzte empfangene Puls aufgezeichnet, so dass sich daraus ein Maß für die Höhe der Vegetation ableiten lässt.

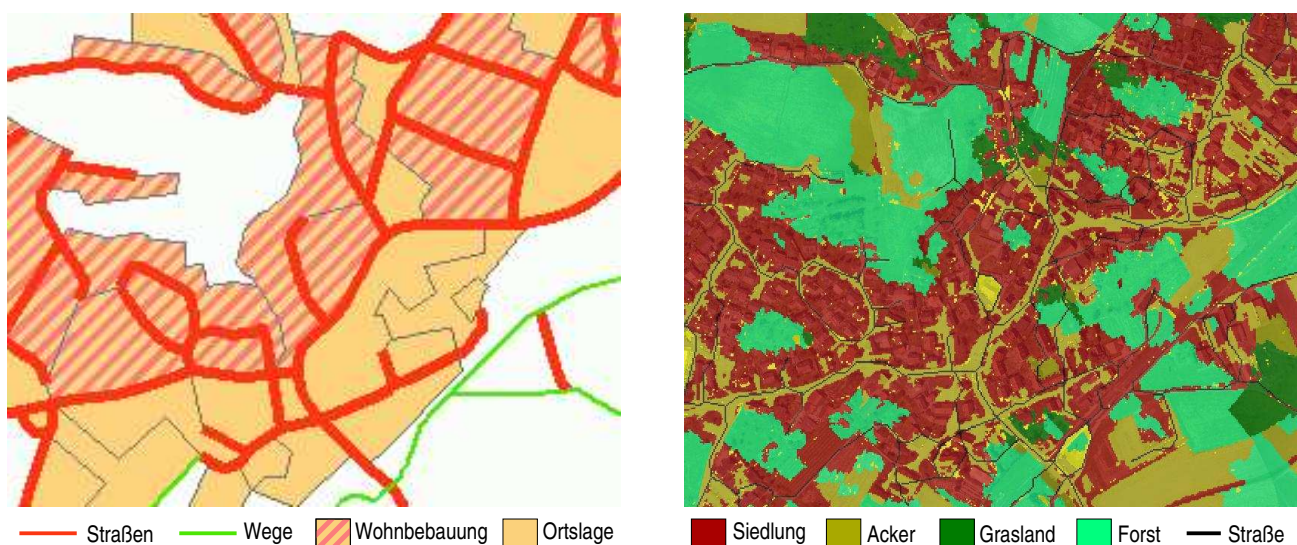


Abbildung 5.11: *Szene Ravensburg: links ATKIS, rechts Interpretation.*

Abbildung 5.11 zeigt links die dem ATKIS für diese Szene entnommenen Flächen für Wohnbebauung und Ortslage. Rechts ist das Ergebnis der automatischen Interpretation zu sehen. Hierbei lässt sich feststellen, dass hinsichtlich der Form der Ortslage und der automatisch detektierten Siedlungen eine hohe Übereinstimmung besteht. Jedoch ist auch zu erkennen, dass der Begriff *Ortslage* offensichtlich administrativ zu verstehen ist und keineswegs die Landnutzung klassifiziert, da einige Flächen innerhalb der *Ortslage* eindeutig landwirtschaftlich genutzt werden. Weiterhin scheint das ATKIS hinsichtlich der mit *Wohnbebauung* ausgewiesenen Flächen nicht aktuell zu sein, da in vielen Bereichen der Ortslage inzwischen Wohnhäuser eindeutig detektiert wurden.

5.3 Beispiel zur satellitengestützten Landnutzungsanalyse

Die Wechselwirkung zwischen Entwicklungen im Umweltbereich und Aktivitäten in der Landwirtschaft und Industrie ist ein wesentliches Thema der Agenda 21. Um die nachhaltige Nutzung natürlicher Ressourcen zu fördern, ist eine koordinierte Überwachung der Umwelt und eine systematische Umweltplanung unerlässlich. Eine kontinuierliche satellitengestützte Erfassung der Erdoberfläche ist ein hilfreicher Baustein, diese Aufgaben zu bewältigen. In der Regel ist jedoch nicht genügend qualifiziertes Personal verfügbar, um die Fernerkundungsdaten auszuwerten. Der Einsatz eines automatischen Interpretationssystems ermöglicht es, Art und Umfang von Veränderungen in der Landnutzung zu erkennen.

In [76] wird die Auswertung von Satellitenbildern aus einem ca. 850km^2 großen Gebiet aus dem Raum Rio de Janeiro in Brasilien mit der Unterstützung von GEOAIDA beschrieben. Die Einrichtung des Interpretationssystems erfolgt dabei durch Experten, die das System nach den Vorgaben eines qualifizierten Bildanalysten konfigurieren. Im Regelbetrieb soll das System später von eingewiesenem Personal ohne spezielle Kenntnisse in der Bildinterpretation bedient werden. Aufgabe des Systems ist es, Änderungen in der Landnutzung zu erkennen und zu interpretieren. Die betreffenden Gebiete können dann von Experten genauer untersucht werden.

Als Eingangsdaten stehen dreibändige (rot, grün und nahes Infrarot) Bilder des Satelliten SPOT XS4 mit einer räumlichen Auflösung von 20m zur Verfügung (s. Abbildung 5.12). Zur Segmentierung der Daten werden zwei verschiedene Bildverarbeitungsoperatoren eingesetzt. Der erste Operator nutzt die spektralabhängige, unterschiedlich starke Reflektion verschiedener Objekte. So reflektiert gesunde Vegetation besonders stark im Grün- und Infrarotband, wohingegen der Rotkanal die Bestimmung unterschiedlicher Boden und Pflanzenarten aufgrund ihres Mineralgehaltes erlaubt. Als zweiter Operator wird das in Abschnitt 5.1.2 beschriebene Textursegmentierungsverfahren verwendet, dass zuvor auf die unterschiedlichen Objekttypen angelernt wurde. Zudem wurde bekanntes Wissen über Straßen- und Flussläufe in den Interpretationsprozess

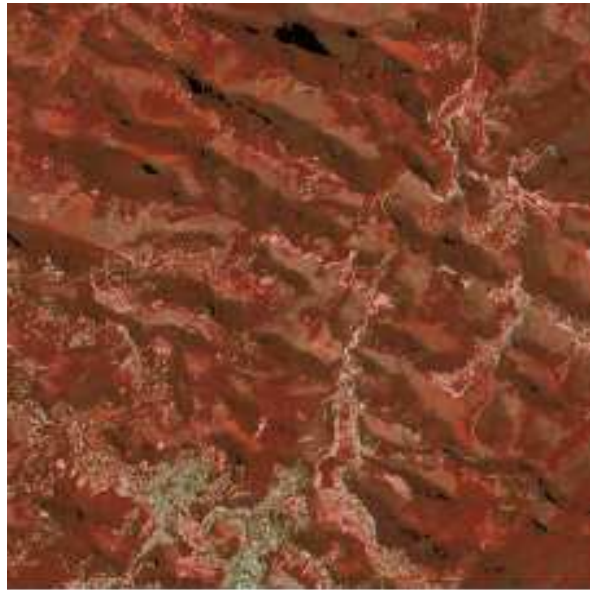


Abbildung 5.12: 3 Bänder (rot, grün und nahes Infrarot) eines Spot XS Satellitenbildes in Falschfarbendarstellung.

einggebracht.

Um das Ergebnis der automatischen Interpretation beurteilen zu können, wurde eine Referenzkarte vom menschlichen Experten generiert. Abbildung 5.13 zeigt links die Referenzkarte und rechts das Ergebnis der automatischen Interpretation. Dabei zeigen sich gute Übereinstimmungen von über 90% bei den Klassen *Felsen* und *Wald*. Die Fehlklassifikation für *Weideland* liegt mit 55% recht hoch, da *Weideland* in Schattenbereichen irrtümlich als *Wald* klassifiziert wurde. Die Einbeziehung von Höheninformationen kann hier helfen Schattenbereiche im Vorfeld zu detektieren. Dünn besiedelte städtische Gebiete bereiten ebenfalls Probleme, da hier die Texturinformation zu schwach ist, andererseits die Bildauflösung nicht ausreicht um diese Bereiche über die Detektion von Gebäuden strukturell zu klassifizieren. So wurden 13% der städtischen Gebiete irrtümlich als *Weideland*, 33% als *Wald* klassifiziert. Bei diesen Ergebnissen ist allerdings zu berücksichtigen, dass der Bildanalyst oftmals weiteres Vorwissen, das nicht dem Bild zu entnehmen ist, wie z.B. Ortskenntnis in die Interpretation mit einbringt bzw. unterschiedliche Bildanalysten zu unterschiedlichen Ergebnissen kommen.

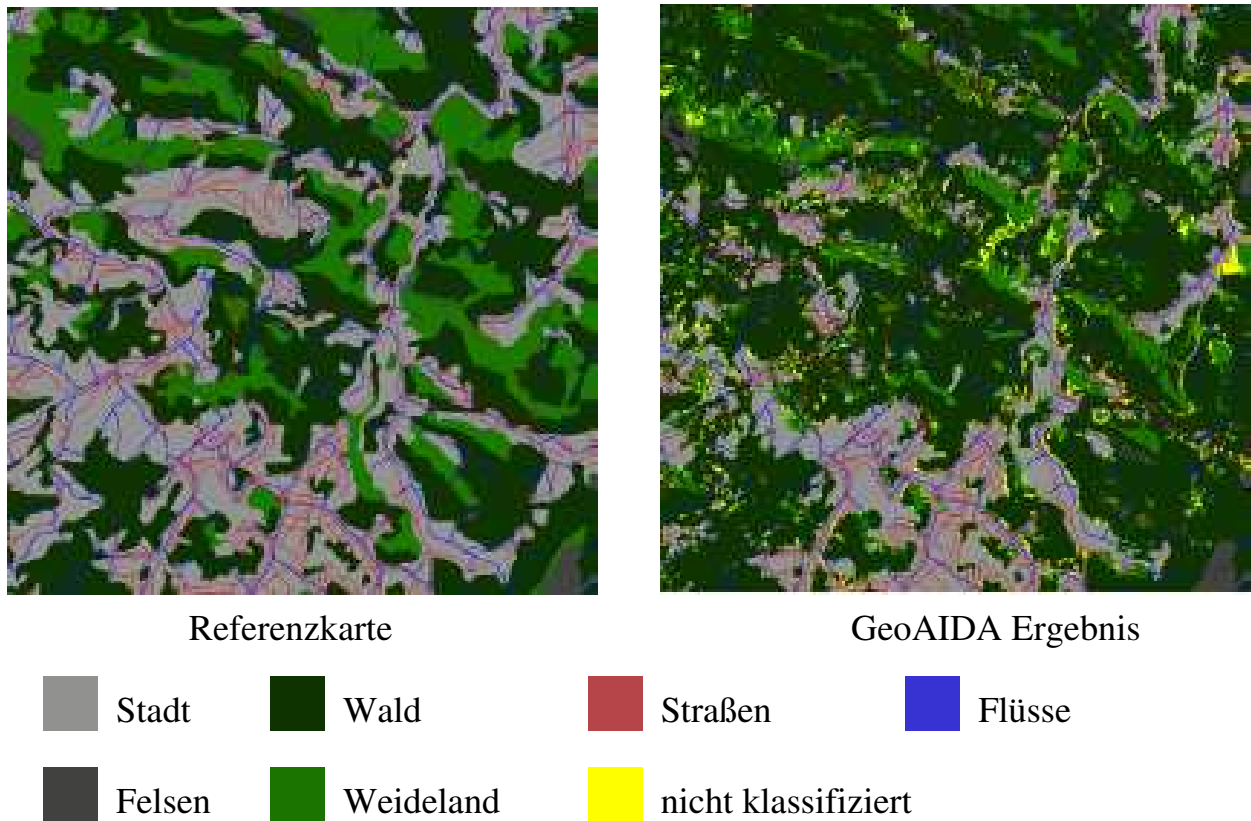


Abbildung 5.13: *Manuelle und automatische Interpretation eines Satellitenbildes.*

Durch die leichte Integrierbarkeit externer Bildverarbeitungsprogramme in GEOAIDA konnte der vorhandene, auf spektrale Eigenschaften der Objekte basierende Segmentierungoperator schnell in das System integriert und mit dem völlig unabhängig von diesem Projekt entwickelten Texturoperator kombiniert werden. Zudem ermöglicht die Auswertung von Nachbarschaftsbeziehungen in Kombination mit den vorhandenen Informationen über Straßen und Flüsse eine weitere Verbesserung der Klassifikation.

6 Bewertung und Ausblick

Die in den Abschnitten 5.2 und 5.3 vorgestellten Anwendungsbeispiele belegen, dass die in dieser Arbeit entwickelte Architektur die Aufgabe, die Interpretation von Fernerkundungsdaten zu steuern, erfüllen kann. Im Gegensatz zu bisherigen wissensbasierten Systemen ist auch die Analyse großer Szenen effizient möglich. Dies liegt zum einen in der hierarchischen Analysestrategie begründet, welche die Suchbereiche für bestimmte Objektklassen frühzeitig einschränken kann. Zum anderen kommt eine auf Parallelisierbarkeit der Analyseschritte optimierte Implementierung hinzu, wodurch eine effiziente Nutzung von Rechnerressourcen erreicht wird. Das offene Konzept für die Anbindung externer Bildverarbeitungs- und Bewertungsoperatoren erleichtert zudem das Erstellen komplexer Anwendungen im Team, insbesondere da bestehende Teillösungen leicht integrierbar sind.

Trotz dieser positiven Ergebnisse ist die Entwicklung der im Rahmen dieser Arbeit entstandenen Architektur nicht als abgeschlossen anzusehen. Als mögliche Weiterentwicklungsrichtungen sind hier der Entwurf einer Multihierarchieanalyse, die automatische Operatorenauswahl, die quantitative Qualitätsbewertung der mit der automatischen Analyse erzielten Ergebnisse sowie die Ausnutzung multitemporaler Datenbestände zu nennen.

6.1 Multihierarchieanalyse

Als Modellwissen wird in GEOAIDA eine Hierarchie der in einer Szene zu erwartenden Objekte verwendet. Dies entspricht einer Ansicht auf die Szene. In der Regel gibt es aber mehrere Sichten, deren Kombination erst das ergibt, was man gemeinhin als Weltbild bezeichnet. So kann beispielsweise eine Straße

Bestandteil einer Ortslage sein. Gleichzeitig ist sie aber auch Bestandteil eines Straßennetzes.

In einem Multihierarchiesystem würden mehrere Modellhierarchien parallel aufgebaut. Die Analyse würde parallel in allen Hierarchien gestartet. Sind in dem oben genannten Beispiel die Straßen in der Straßennetzhierarchie nur über den Kontext der Landnutzungshierarchie zu detektieren, so würde in der Straßennetzhierarchie Hypothesen bis zur Ebene der Straßenabschnitte generiert. In der Landnutzungshierarchie würde die Szene in die verschiedenen Objektklassen aufgeteilt, bis dort schließlich auch die Ebene mit den Straßenabschnitten erreicht ist. Sobald dort die Hypothesen für Straßenabschnitte kontextspezifisch generiert sind, kann in der Straßennetzhierarchie fortgefahren werden, entweder mit der Suche nach weiteren untergeordneten Objektklassen oder mit der strukturellen Analyse der Straßenabschnitte, um sie zu übergeordneten Straßenzügen zusammenzufassen.

Zu behandelnde Probleme sind zum einen das Design der komplexeren Analyse, die der jetzigen übergeordnet wäre, zum anderen aber auch Fragen der Visualisierung. Dem Benutzer müssten einerseits die einzelnen Sichten auf die Szene ermöglicht, andererseits aber auch die Verknüpfungen zwischen den Hierarchien angezeigt werden.

6.2 Automatische Operatorenauswahl

Die Auswahl der Segmentierungs- und Klassifizierungsoperatoren geschieht zur Zeit manuell und setzt Kenntnisse beim Anwender über die Verfügbarkeit und grundlegende Funktionalität der Operatoren voraus. Im Prinzip hängt die Auswahl des einzusetzenden Operators jedoch von der gesuchten Objektklasse und dem zur Verfügung stehenden Datenmaterial ab, so dass eine Auswahl automatisch erfolgen könnte. Hierdurch könnte das System sowohl auf das Fehlen als auch auf das Hinzukommen einer neuen Datenquelle reagieren und die Interpretation selbstständig den geänderten Randbedingungen optimal anpassen.

Eine Erweiterung des Systems in dieser Richtung ist im Prinzip recht einfach. Den Operatoren ist eine detailliertere Beschreibung der zulässigen Arten von

Daten hinzuzufügen, wie sie jetzt schon in Ansätzen vorhanden ist. Des Weiteren werden sie um ein Qualitätsmaß ergänzt, das ihre Eignung bezüglich der Detektion einer bestimmten Objektklasse beschreibt. Beim Design der Operatoren ist darauf zu achten, dass Werte für Parameter möglichst aus dem Datenmaterial selbst ermittelt werden, damit eine manuelle Konfiguration überflüssig wird. Eventuell eignen sich für die Realisierung agentenbasierte Systeme, wie sie auch in [25] als eine Form der Wissensnutzung beschrieben sind.

6.3 Quantitative Qualitätsbewertung

Bei der Automatisierung von bisher manuell durchgeführten Verfahren stellt sich natürlich die Frage nach der Qualität des automatischen Verfahrens gegenüber der manuellen Vorgehensweise. Qualitativ kann dies durch die bereits implementierte Überlagerung der Ergebniskarten mit beliebigen anderen Datensätzen geschehen. Quantitativ wäre hierzu eine Überlagerung zweier Objekthierarchien notwendig, die auf dem gleichen Modellnetz basieren, um dann in jeder Ebene, die Schnittmenge gleicher Klassen im Verhältnis zur Referenzmenge zu bestimmen.

6.4 Auswertung multitemporaler Datenbestände

Multitemporale Datensätze werden in der Regel ausgewertet, um Veränderungen in den betrachteten Szenen festzustellen (engl. Change detection). Anwendungabhängig können dabei Veränderungen an künstlichen Objekten wie z.B. der Stadtentwicklung oder Veränderungen an der Vegetation z.B. durch einen veränderten Wasserhaushalt eine Rolle spielen.

Pixelbasierter Vergleich für die Fernerkundung ungeeignet Eine nahe liegende Lösung zur Detektion von Veränderungen in multitemporalen Bildsequenzen ist der pixelbasierte Vergleich. Er setzt jedoch eine exakte pixelgenaue Positionierung und konstante Belichtungsverhältnisse voraus, wie sie bei der

Erfassung von Fernerkundungsdaten nicht gegeben sind. Der pixelbasierte Vergleich eignet sich daher primär nur für Überwachungsaufgaben in geschlossenen Räumen, da hier konstante Beleuchtungsverhältnisse erzeugt werden können. Zudem ist eine pixelgenaue Positionierung durch Montage der Kamera auf einem Stativ gegeben, so dass letztlich als einzige Störgröße das Bildrauschen bleibt. Werden die erhöhte Anforderung an die Konstanz der Beleuchtung und der Bildposition erfüllt, stellt der pixelbasierte Vergleich eine einfach zu realisierende Möglichkeit zur automatischen Detektion von Änderungen dar. Insbesondere eignet sich das Verfahren auch zur halbautomatischen Überwachung, indem Bilder nur bei großen Änderungen gesichert werden. Der Auswerter muss dann nur die Bilder mit Bewegung überprüfen. Für den Einsatz in Fernerkundungsdaten ist der pixelbasierte Vergleich wegen wechselnder Beleuchtungsverhältnisse und ungenauer Positionierung des Bildmaterials ungeeignet.

Vergleich über lokale statistische Werte Möchte man einen Vergleich auf Bildebene durchführen, so bietet sich der Vergleich auf Basis lokaler statistischer Werte an, d.h. es werden statistische Werte innerhalb eines sich über das Bild bewegenden Fensters berechnet und diese miteinander verglichen. Als statistische Größen sind z.B. Mittelwert (Moment 1. Ordnung), Medianwert und Varianz (Moment 2. Ordnung) in Betracht zu ziehen. Dabei sind prinzipiell mittelwertfreie Größen wie die Varianz zu bevorzugen, da sie sich toleranter gegenüber sich ändernden Beleuchtungsverhältnissen verhalten. Zwar lassen sich mit derartigen Verfahren Wahrscheinlichkeiten für die Veränderung von Regionen bestimmen, jedoch können keine Aussagen über die Art der Veränderung gemacht werden, da den Bildbereichen keine Semantik zugeordnet wird.

Struktureller Vergleich Beim strukturellen Vergleich werden die multitemporalen Datensätze zunächst interpretiert. Der Vergleich erfolgt dann auf der Bedeutungsebene. Dies setzt voraus, dass unveränderte Bildinhalte auch gleich klassifiziert werden. In [42] wird daher vorgeschlagen, Übergangswahrscheinlichkeiten zwischen den Klassen einzubringen, um so die Detektion von Veränderungen in plausible Richtungen zu lenken. So ist es beispielsweise unwahrscheinlich, dass auf einem Ackerland innerhalb eines Jahres

ein dichter Forst entsteht, wohingegen die Entstehung eines neuen Wohngebietes auf einem Ackerland bei schon benachbarter Bebauung ziemlich wahrscheinlich ist. Eine korrekte initiale Klassifizierung wird bei diesem Verfahren allerdings umso wichtiger. Zudem kann eine einmalige Fehlklassifizierung zu einem Drift in den Folgeklassifizierungen führen.

Einsatz multitemporaler Datensätze in GeoAIDA Grundsätzlich ist der Einsatz multitemporaler Datensätze in GEOAIDA an drei verschiedenen Stellen denkbar: zur Erwartungssteuerung und zur Parameteradaption in Top-Down-Phase sowie zur Änderungsdetektion während des Bottom-Up-Laufs. Bei allen hier vorgestellten Integrationsmöglichkeiten multitemporaler Datensätze ist ein Vorhalten älterer Bildsequenzen nicht erforderlich.

Erwartungssteuerung Eine Erwartungssteuerung ließe sich beispielsweise über eine veränderte Bewertung der durch einen Top-Down-Operator entstandenen Objektliste unter Zuhilfenahme einer früheren Interpretation und Übergangswahrscheinlichkeiten erreichen. Die Übergangswahrscheinlichkeiten lassen sich dabei als Attribute der Objekte integrieren. Ergänzt werden müsste diese Evaluation in der Auswertung der Top-Down-Ergebnisse.

Parameteradaption Mit der Parameteradaption von Bildverarbeitungsalgorithmen im industriellen Umfeld beschäftigt sich [77]. Inwieweit diese Erkenntnisse auf die Auswertung von Fernerkundungsdaten übertragbar sind, müsste zunächst genauer untersucht werden, da die Randbedingungen in der industriellen Bildverarbeitung in der Regel wesentlich konstanter sind. Grundsätzlich müsste die Parameteradaption in die Top-Down-Operatoren integriert werden, indem diesen die Interpretation eines Teilbereichs vorgegeben wird. Innerhalb des Top-Down-Operators müsste dann ein Regelkreis in Gang gesetzt werden, der die Parameter solange anpasst, bis mit dem neuen Datenmaterial für den vorgegeben Bildausschnitt die gleiche Interpretation erreicht wird, in der Hoffnung, dass sich für die restlichen Bildbereiche mit dieser Parametereinstellung ebenfalls korrekte Ergebnisse erzielen lassen.

Änderungsdetektion Grundsätzlich ist eine Änderungsdetektion während der Top-Down- oder Bottom-Up-Phase möglich. Eine Detektion während des Top-Down-Laufs hätte etwa den im folgenden skizzierten Ablauf:

Zunächst werden die übergeordneten Regionen z.B. über lokale statistische Werte verglichen. Werden Änderungen detektiert, so werden in diesen Regionen die Detailregionen verglichen. Da es unwahrscheinlich ist, dass Regionen in zwei unterschiedlichen Bildern exakt übereinstimmen, sind Toleranzen beim Vergleich zuzulassen. Diese Toleranzen können relativ zur Bildgröße sein, mit dem Nachteil, dass dann sehr kleine Regionen nicht erkannt werden. Als Beispiel sei hier z.B. der Bauernhof genannt, der um eine Scheune auf einem bisherigen Feld erweitert wird. Im Verhältnis zur Feldgröße erscheint der Neubau einer Scheune als sehr kleine Änderung, die eventuell nicht erkannt wird. Im Verhältnis zu Gehöftgröße ist die Region, welche die Scheune belegt, nicht zu vernachlässigen.

Werden hingegen absolute Toleranzen zugelassen, stellt sich die Frage, wie hoch diese zulässigen Abweichungen sein dürfen. Wurde in einem großen Wald ein Baum gefällt, so ist diese Änderung nur von wenig Interesse. Wird jedoch ein Haus darin gebaut, so ist dies eine Änderung der Landnutzung.

Besser lassen sich Veränderungen in der Szene während der Bottom-Up-Phase detektieren, da hier beginnend mit den kleinsten Strukturen immer größere zusammenhängende Objekte gebildet werden. Veränderungen werden aber meistens zunächst in den kleinen Strukturen sichtbar, wohingegen es sich bei kleinen Grenzverschiebungen in höheren Ebenen der Objekthierarchie eher um Interpretationsungenauigkeiten handelt.

Visualisierung von Veränderungen Neben der Detektion von Veränderungen müssen diese dem Benutzer auch in geeigneter Weise angezeigt werden. Es gilt deshalb, den Vergleich zweier Hierarchien zu visualisieren und die Aufmerksamkeit auf die veränderten Regionen zu lenken. Dabei stellt sich die Frage der Relevanz von Veränderungen, die in der Regel kontextabhängig zu beantworten ist. Werden in europäischen Waldgebieten Bäume gefällt, so geschieht dies in der Regel aus forstwirtschaftlichen Gründen. In den Regenwäldern Südamerikas hingegen kann dies ein Hinweis auf eine beginnende illegale Zerstörung sein.

7 Zusammenfassung

Inhalt der vorliegenden Arbeit ist der Entwurf einer Architektur eines wissensbasierten Systems zur automatischen Interpretation von Fernerkundungsdaten. Das System steuert den Interpretationsablauf von Luft- und Satellitenbildern und generiert eine symbolische Szenenbeschreibung, die den Objekten im Bild eine Semantik zuordnet. Die symbolische Szenenbeschreibung wird in Form eines Baumes angelegt. Jeder Knoten korrespondiert mit einem Objekt im Datenmaterial. Die genaue Lage jedes Objekts wird mit einem objektspezifischen Identifikationswert in Labelbildern verzeichnet. Mit Hilfe des Szenenbaumes und der Labelbilder lassen sich Karten unterschiedlichen Detaillierungsgrades erzeugen.

Die Auswertung von Fernerkundungsdaten geschieht für die vielfältigsten Anwendungsgebiete heutzutage zumeist manuell. Die Einsatzgebiete dienen dabei sowohl zivilen als auch militärischen Zwecken. Zu den zivilen Aufgaben gehören die Kartenerstellung, die Städteplanung, die Landschaftsnutzung, der Umweltschutz, der Katastrophenschutz, das Erstellen, Verifizieren und Aktualisieren von Geoinformationssystemen sowie die 3D-Stadtmodellierung. Hinzu kommen die militärischen Aufgaben der Luftaufklärung und Kartenerstellung. Mit steigender Zahl der Aufgaben und dem Wunsch nach ständig aktuellen Informationen gewinnt auch der Einsatz von automatischen Verfahren zur Interpretation der Fernerkundungsdaten an Bedeutung. Diesem Trend trägt das hier vorgestellte Verfahren Rechnung, indem es versucht, bereits bekannte und erprobte Verfahren zur Interpretation bestimmter Objektklassen zu integrieren und zu einem, für den praktischen Einsatz geeigneten Gesamtsystem auszubauen.

Neben den Fernerkundungsdaten selbst benötigt das System für die Interpretation der Daten ein Modell der zu erwartenden Objekte. Dieses Modell wird in Form eines Baumes angelegt, in dem untergeordnete Objekte immer vollständig

in übergeordneten Objekten enthalten sind. Diese Objekte werden als Knoten des Baumes bezeichnet. Jedem der Knoten kann ein Bildverarbeitungsoperator zugeordnet werden, der in der Lage ist, Objekte diesen Typs aus dem bereitgestellten Bildmaterial zu extrahieren, oder zumindest die Fähigkeit besitzt, die möglichen Suchbereiche für Objekte diesen Typs vorzugeben.

Der Aufbau dieser Objekthierarchie spiegelt damit die Vorgehensweise eines Betrachters wieder, der sich einer Szene nähert und dabei zunächst übergeordnete Objekte erkennt. Je näher er der Szene kommt, desto mehr Struktur erhält die Szene und die Teilobjekte werden sichtbar. Manche Objekte sind dabei erst über die Struktur ihrer Teilobjekte identifizierbar, andere wiederum lassen sich nur im übergeordneten Kontext erkennen.

Die Analysestrategie des hier vorgestellten Systems arbeitet nach demselben Prinzip. Zunächst wird versucht, die Szene in immer kleinere Teilbereiche zu zerlegen. Dabei kommen objektspezifische Bildverarbeitungsoperatoren zum Einsatz, um Suchbereiche für untergeordnete Objekte einzuschränken. Für jeden der gefundenen Suchbereiche wird ein Knoten in einem sich analog zum Modellnetz aufbauenden, hypothetischen Instanzennetz angelegt. Ist kein objektspezifischer Operator verfügbar, wird der komplette Suchbereich der übergeordneten Instanz durchgereicht. Sind die Blattknoten des Modellnetzes erreicht, erfolgt eine Verifikation des hypothetischen Instanzennetzes beginnend mit den Blattknoten. Dazu wird ein objektspezifischer Bewertungsoperator aktiviert, der die Struktur seiner untergeordneten Teilobjekte untersucht, einzelne Teilobjekte verwirft oder modifiziert und die Teilobjekte zu größeren Objekten gruppiert. Die Analyse wird auf diese Art fortgesetzt bis der Wurzelknoten erreicht ist. Nach Abschluss der Interpretation werden mit Hilfe des Instanzennetzes die einzelnen, während der Interpretation entstandenen Labelbilder in einer Karte so codiert abgelegt, dass sich mit Hilfe des Instanzen- und Modellnetzes interaktiv Karten in verschiedenen Detaillierungsgraden erzeugen lassen. Dabei sind Kombination mehrerer Detaillierungsgrade in verschiedenen Bereichen der Karte ebenso möglich wie auch der Zugriff auf die Objektattribute durch Auswahl in der Karte.

Die bisherigen Ansätze zur Auswertung von Fernerkundungsdaten lassen sich in zwei Kategorien einteilen. Zum einen wird an der Entwicklung von Algorithmen

gearbeitet, die wenige Objektklassen in einem bestimmten Kontext detektieren können. Zum anderen gibt es integrierte Systeme, die sich an der Interpretation ganzer Szenen versuchen. Die vielversprechendsten Systeme dieser zweiten Kategorie halten das Wissen über die Szene in semantischen Netzen vor. Diese semantischen Netze entsprechen einem Modell der Szene. Jeder Knoten des Netzes entspricht einem möglicherweise in der Szene vorhandenen Objekt. Die Beziehungen der Objekte untereinander werden in Kanten modelliert, welche die Knoten verbinden. Um eine regelbasierte Analyse zu ermöglichen, werden den Kanten bestimmte Bedeutungen zugewiesen. Einige Systeme erlauben nur die Beschreibung hierarchischer Beziehungen wie *part-of*, *is-a* etc. Andere führen topologische Beziehungen ein, um beispielsweise die Nachbarschaft von Objekten zu beschreiben. All diesen Verfahren ist gemein, dass die eigentliche Bildverarbeitung nur auf der Ebene der Blattknoten angebunden ist. Die Interpretation erfolgt durch strukturelle Analyse der aus den Bildern gewonnenen primitiven Objekte. Wegen der großen Zahl der in Fernerkundungsdaten vorhandenen Bildprimitive und der daraus resultierenden vielen Kombinationsmöglichkeiten sind diese Verfahren für die Interpretation von Fernerkundungsdaten nur bedingt geeignet.

Das in dieser Arbeit vorgestellte System verwendet ebenfalls ein semantisches Netz zur Beschreibung des Modellwissens. Das semantische Netz ist streng hierarchisch und wird gleichzeitig als Steuerungsnetz für die Analyse eingesetzt. Den Kanten kommt keine besondere Bedeutung zu. Sie verbinden lediglich Objekte unterschiedlicher Hierarchieebenen. Ihre Bedeutung ist in Bewertungsoperatoren ausgelagert, die je nach spezieller Anforderung alle möglichen Kantentypen nachbilden können. So sind *part-of*-Beziehungen genauso modellierbar wie topologische Relationen der Objekte einer Hierarchieebene.

Die Datenanbindung erfolgt in jedem Knoten des semantischen Netzes, sofern ein Verfahren bekannt ist, das den Suchbereich für Objekte des betreffenden Knotentyps einschränken kann. Diese Verfahren sind dabei nicht auf die Bildverarbeitung beschränkt. Gleichwertig lassen sich auch beispielsweise Datenbankabfragen an ein GIS integrieren. Durch das frühzeitige Einschränken des Suchbereichs auf höheren Objektebenen ist ein vollständiges Durchsuchen aller Kombinationen von Bildprimitiven nach passenden Lösungen nicht nötig. Hierdurch wird eine Effizienzsteigerung erreicht.

Die an dem hierarchischen Modell orientierte Analysestrategie teilt die Ausgangsszene in immer kleinere Teilbereiche, die unabhängig voneinander behandelt werden können. Hierdurch entsteht eine hohe Parallelisierbarkeit, die das System für den Einsatz auf Multiprozessorrechnern bzw. in Computerclustern prädestiniert.

Besonderer Wert wurde auf eine einfache und flexible Schnittstelle zur Anbindung externer Datenverarbeitungsprogramme gelegt. Diese werden sowohl zur Segmentierung und Klassifizierung als auch zur Bewertung eingesetzt. Das System ist daher insbesondere für den kooperativen Einsatz in bestehenden Projekten geeignet, um dort die Fusion von Teilergebnissen verschiedener Analyseprozesse zu koordinieren. In Projekten zur Verifikation von ATKIS und in der Auswertung von Satellitenbildern zur Landnutzungsänderung in Waldgebieten Südamerikas hat der hier erarbeitete Ansatz seine universelle Einsetzbarkeit unter Beweis gestellt.

Durch die Lösungsstrategie, Szenen hierarchisch aufzuteilen und Teilprobleme lokal zu behandeln, stellt die hier vorgestellte Architektur einen einfachen, überschaubaren, robusten und effizienten Beitrag zu mehr Praxistauglichkeit von automatischen Interpretationssystemen in der Fernerkundung dar.

Glossar

A

A* Mit A* wird ein Suchalgorithmus bezeichnet, der aus einer Lösungsmenge die optimale Lösung sucht, indem er die jeweils vielversprechendste weiter entwickelt. Die Auswahl der vielversprechendsten Lösung geschieht über eine Abschätzung der Gesamtkosten, die sich aus den bereits aufgelaufenen Kosten und den zu erwartenden Restkosten zusammensetzen. Die Suche ist beendet, sobald ein Lösungsweg sein Ziel erreicht hat. (S. xvi)

American Standard Code for Information Interchange (ASCII) Codierung für die Übertragung von einfachen Zeichenfolgen, S. 41.

Amtlich Topographisch-Kartographisches Informationssystem (ATKIS) amtliches Geoinformationssystem der Bundesrepublik Deutschland, S. 30.

B

Balloon-Help Als Balloon-Help bezeichnet man eine Hilfestellung in Benutzeroberflächen, die einem Ballon gleich erscheint, wenn der Mauszeiger auf einem Element der Benutzeroberfläche positioniert wird. (S. 77)

Bottom-Up-Operator Der Bottom-Down-Operator erhält eine Liste von Objekten mit dazugehörigen Regionen. Diese muss er bewerten, Konflikte zwischen Objekten auflösen und gruppieren. (S. 56)

Boundingbox Die Boundingbox bezeichnet das zu den Koordinatenachsen parallel ausgerichtete, umschließende Rechteck einer Region.

Byteorder Daten, die einen Wertebereich von [0..255] überschreiten, lassen sich nur in mehr als einem Byte ablegen. Sollen die Daten in einer Datei abgelegt werden, muss entschieden werden, ob zunächst die höherwertigen Bytes (Big-Endian-System) oder die niederwertigen Bytes (Little-Endian-System) gespeichert werden. In welcher Weise Datentypen gespeichert werden, hängt davon ab, wie der Prozessor die Daten verarbeitet. So handelt es sich bei Intel-Prozessoren z.B. Little-Endian-Systeme, wohingegen die PowerPC-Architektur, aber auch viele RISC-Prozessoren Big-Endian verwenden. Schwierigkeiten ergeben sich bei Übergabe der Daten von Systemen mit unterschiedlicher Byteorder. Nur wenn die Größe der Datentypen bekannt ist, können die Daten korrekt gewandelt werden. (S. 41)

C

Cache Mit Cache bezeichnet man einen Datenspeicher mit kurzen Zugriffszeiten, der einem langsameren Datenspeicher vorgeschaltet wird. Häufig benötigte Daten werden im Cache vorgehalten und können bei Bedarf schnell zur Verfügung gestellt werden. Sind die Daten hingegen nicht im Cache verfügbar, muss auf das langsame Hauptspeichermedium zugegriffen werden. Die Daten werden in den Cache übertragen, um sie bei einem erneuten Zugriff schnell bereit stellen zu können. Bei jedem Zugriff auf den Cache wird die Bewertung der entsprechenden Daten erhöht. Ist der Cache gefüllt, werden die Daten mit der niedrigsten Bewertung entfernt. (S. 67)

Compiler Ein Compiler dient dazu, Programmtext in eine hardwarenahe Beschreibung zu wandeln, die direkt vom Prozessor ausgeführt werden kann (s. Interpreter). (S. 129)

CSV CSV ist ein Farbraum, in dem Farben aus den Komponenten Farbe (Color), Sättigung (Saturation) und Helligkeitswert (Value) dargestellt

werden. Dieser Farbraum ist für Segmentierungsaufgaben besser geeignet, da Grauwerte und Farben getrennt vorliegen. (S. 92)

D

Digitales Geländemodell (DGM) Ein Digitales Geländemodell beschreibt das Höhenprofil einer Szene. Es gibt verschiedene Möglichkeiten ein DGM zu erzeugen, wie z.B. die Berechnung aus Stereodaten, das Scannen der Oberfläche mit einem Laser oder die Vermessung durch Ortsbegehung. (S. 96)

E

eXtended Markup Language (XML) In XML können attributierte Objekt-hierarchien abgelegt werden. XML ist eine Sprache, die man als Verallgemeinerung von Hypertext Markup Language (HTML) bezeichnen kann. Während HTML einen festen Satz von Objekten kennt, wird in XML die Bedeutung der Objekte in einer Hilfsdatei definiert. GEOAIDA verzichtet auf diese Hilfsdateien. Die Bedeutung der Objekte ist kontextspezifisch festgelegt. (S. 41)

F

FIFO FIFO steht für *First In First Out* und bedeutet, dass das Datum, welches zuerst in den Datencontainer eingestellt wurde, als erstes den Datencontainer wieder verlässt. Dies entspricht der Arbeitsweise eines Fließbandes. Der Gegenstand, der zuerst auf das Fließband gelegt wird, erreicht als erster das Ende (vgl. LIFO). (S. 131)

Filter Ein Filter ist ein Programm, das Daten einer Transformation unterzieht, um die Daten bezüglich bestimmter Informationen zugänglicher zu machen. (S. 87)

Finite State Machine (FSM) Als Finite State Machine bezeichnet man einen Automaten der nur eine begrenzte Anzahl von Zuständen annehmen kann. Diese Zustände werden in der Regel in einer vorgegebenen Reihenfolge durchlaufen, wobei externe Ereignisse die Abarbeitungsreihenfolge verändern können. (S. 67)

Footprint Als Footprint bezeichnet man die 2D-Projektion eines Objekts in eine Ebene. (S. 58)

G

GeoAIDA GEOAIDA steht für Geo Automatic Image Data Analyser. Es bezeichnet im allgemeinen ein Projekt mit Werkzeugen zur Auswertung von Fernerkundungsdaten. Im speziellen steht GEOAIDA für ein spezielles Verfahren mit Hilfe von Modellwissen, Fernerkundungsdaten automatisch zu interpretieren. (S. 34)

Geoinformationssystem (GIS) Ein Geo-Information-System bezeichnet ein System, das lokalisierte Informationen zur Verfügung stellt. Üblicherweise werden diese Informationen in Datenbanken vorgehalten, die um die Möglichkeit der spatialen Abfrage erweitert wurden. (S. 30)

Geokoordinaten Geokoordinaten ist eine Kurzform für ein geodätisches Koordinatensystem. In einem geodätischen Koordinatensystem lassen sich Positionen auf der Erdoberfläche eindeutig beschreiben. (S. 41)

H

holistisch Das Wort *holistisch* bedeutet soviel wie *ganzheitlich*. Im System GEOAIDA wird das Wort im Zusammenhang mit Operatoren benutzt. Ein holistischer Operator ist ein Programm, das die Detektion einer oder mehrerer Objektklassen auf „magische“ Weise erlaubt, d.h. durch eine nicht näher spezifizierte, ganzheitliche Betrachtung der

Eingangsdaten, mit Ziel die Zahl der möglichen Hypothesen einzuschränken. (S. 54)

Hypertext Markup Language (HTML) Die Hypertext Markup Language dient dazu Texte strukturell zu beschreiben. (S. 127)

I

Instanzennetz Das Instanzennetz ist das Ergebnis der Anwendung des Modellnetzes auf eine konkrete Szene. Objekte des Modells, die in der Szene vorhanden sind, werden instanziiert. Ein Modellobjekt kann einfach, mehrfach oder gar nicht im Instanzennetz instanziiert sein. Während der Analyse werden die Knoten des Instanzennetzes als Hypothesen bezeichnet. Erst mit der Verifikation des Szeneknotens wird das gesamte Instanzennetz bestätigt. (S. 51)

interaktive Karte Eine interaktive Karte ist das Ergebnis der Analyse mit GEOAIDA. Sie besteht aus einem Modellnetz, einem Instanzennetz und einem zweidimensionalen Bild, in dem die Karten in verschiedenen Details kodiert abgelegt sind, sowie einem Programm, welches mit Hilfe der Daten die vom Benutzer gewünschte Karte erstellt. (S. 51)

Interpreter Ein Interpreter führt ein Programm aus, indem er den Programmtext interpretiert und direkt zur Ausführung bringt (s. Compiler). (S. 88)

K

Konzeptnetz s. semantische Netz.

L

Labelbild Ein Labelbild dient dazu, die Lage und Ausdehnung von Objekten pixelgenau zu beschreiben, indem den Pixeln, die innerhalb der

Objektgrenzen liegen, ein Identifikationswert zugewiesen wird.

LIFO LIFO steht für *Last In First Out* und bedeutet, dass das Datum, welches als letztes in den Datencontainer eingestellt wurde, als erstes den Datencontainer wieder verlässt. Dies entspricht der Arbeitsweise eines Stapels. Der Gegenstand, der zuletzt auf den Stapel gelegt wird, muss als erstes wieder entnommen werden (vgl. FIFO). (S. 132)

M

Modellnetz s. semantische Netz, S. 66.

O

Operator Als Operator wird im System GEOAIDA ein eigenständiges Programm bezeichnet, das über eine definierte Schnittstelle spezifische Datenverarbeitungsschritte durchführt. In GEOAIDA wird zwischen Top-Down-Operatoren und Bottom-Up-Operatoren unterschieden. (S. 34)

P

Pipe Als Pipe bezeichnet man eine spezielle Form von Datei, auf die zwei Prozesse gleichzeitig zugreifen: Der eine Prozess schreibt in die Datei, während der andere gleichzeitig daraus liest. Pipes dienen zum Datenaustausch zwischen zwei auf dem selben Computer laufenden Prozessen. Pipes sind normalerweise namenlos, d.h. sie lassen sich nur durch Vererbung zwischen zwei Prozessen aufbauen. Sogenannte Named-Pipes besitzen diesen Nachteil nicht. Sie erscheinen im Dateisystem und können von zwei Prozessen wie normale Dateien geöffnet werden. Weiterführende Informationen sind in [99] zu finden. (S. 49)

Pixel Pixel ist die Kurzform für Picture-Element und beschreibt einen Bildpunkt. (S. 44)

Portable-Any-Map-Format (PAM-Format) Das PAM-Format ist Datenformat zum Ablegen von Bildern. Es besteht aus einem ASCII-Header, der die nachfolgenden Bilddaten beschreibt, und den eigentlichen Daten, die i.d.R. binär abgelegt werden. In seiner ursprünglichen Version unterstützt das Format 8-bit-Daten. In jüngster Zeit wurde das Format um größere Datentypen erweitert. Fließkommazahlen sind bisher jedoch nicht vorgesehen. Unterstützt werden Farbbilder (Pixmap = PPM), Grauwertbilder (Grey = PGM) und Bitmaps (PBM). (S. 42)

Prozess Ein Prozess bezeichnet ein eigenständig ablaufendes Programm, welches im Gegensatz zu einem Thread von anderen gleichzeitig ablaufenden Programmen speichertechnisch gekapselt abläuft. (S. 49)

Q

Queue Eine Queue ist ein FIFO-Datencontainer. (S. 67)

R

Reihenmesskamera (RMK) Reihenmesskameras werden zur Erfassung von Luftbildern verwendet. Das Luftbild wird dabei auf einen Film aufgezeichnet. Es sind Farb-, Grauwert- und Infrarotbilder möglich. (S. 103)

S

Script Eine Script bezeichnet eine Abfolge von Befehlszeilen, die beim Aufruf des Scriptes von einem Interpreter zur Ausführung gebracht werden. (S. 32)

semantische Netz Das semantische Netz beschreibt in GEOAIDA das Wissen über die Szene. Es handelt sich dabei um ein generisches Modell

über die zu erwartenden Objekte. Das Netz wird daher auch als Modellnetz oder Konzeptnetz bezeichnet. GEOAIDA verwendet streng hierarchische Netze, d.h. das Netz hat die Form eines Baums. (S. 51)

Socket Im Gegensatz zu Pipes ist über Sockets der Datenaustausch zwischen Prozessen, die auf verschiedenen Computern laufen, möglich. Dazu wird auf jeder Seite eine Socket geöffnet, über den der Datentransfer abgewickelt wird. Sockets werden z.B. für diverse Netzwerkprotokolle eingesetzt. Weiterführende Informationen sind in [99] zu finden. (S. 49)

spatial = räumlich, S. 128.

Stack Eine Stack ist ein LIFO-Datencontainer. (S. 90)

Standard Template Library (STL) Die Standard Template Library ist eine Sammlung von Template-Datencontainern und darauf anwendbare Template-Funktionen. (S. 44)

Synthetic Aperture Radar (SAR) Synthetic Aperture Radar bezeichnet eine Methode, die Apertur einer Antenne synthetisch zu vergrößern, indem Mikrowellen kohärenter Phase über eine längere Flugstrecke abgestrahlt und empfangen werden. (S. 103)

Szene Die Szene ist immer der oberste Knoten im Modellnetz und im Instanzennetz und bezeichnet die Menge aller Objekte im betrachteten Weltausschnitt.

T

Task Task bezeichnet eine zu erledigende Aufgabe. Im Rechnerkontext wird dies oft gleichgesetzt mit einem Prozess. (S. 66)

Template Unter Template-Klassen und -Funktionen versteht man parametrisierten Programmtext. Templates sind quasi ein Programmmodell, welches erst durch das spätere Setzen der Parameter instanziiert und

damit zu konkretem, compilierbarem Programmcode wird. Die Parameter werden häufig für das Setzen von Datentypen verwendet. Eine Templateklassenbibliothek enthält lediglich Klassen- und Funktionsmodelle. Die Flexibilität der zu Compilezeit möglichen Typisierung wird mit dem Nachteil erkaufte, dass die Bibliothek für sich nicht vorab compilierbar ist. Hierdurch verlängern sich bei der Programmentwicklung die Zeiten für den Compilervorgang. (S. 43)

Thread Die Abarbeitung eines Programms kann in mehreren gleichzeitig laufenden Thread erfolgen, die sich jedoch die Daten teilen. Handelt es sich bei einem Thread um ein eigenständiges Programm, das speichertechnisch von anderen Programmen gekapselt abläuft, so spricht man von einem Prozess. (S. 131)

Timer Ein Timer ist ein Gerät oder Objekt, das regelmäßig ein Signal ausgibt. (S. 67)

Token Als Token bezeichnet man ein Symbol, das stellvertretend für einen bestimmten Sachverhalt oder ein bestimmtes Objekt steht. (S. 91)

Top-Down-Operator Aufgabe eines Top-Down-Operators ist es, aus den ihm zur Verfügung gestellten Daten, eine Liste von attributierten und klassifizierten Regionen zu generieren. (S. 55)

U

Umgekehrt Polnische Notation (UPN) UPN ist eine Postfixnotation, d.h. es werden immer erst die Operanden und dann die darauf anzuwendenden Operatoren angegeben. Das Ergebnis wird an die Stelle des Operators gesetzt und mit der Verarbeitung fortgefahren. (S. 89)

V

visuelles Bild (VIS) Mit VIS wird ein Bild aus dem visuellen Lichtspektrum bezeichnet. Es kann sich dabei um ein Farb- oder Grauwertbild han-

deln. (S. 52)

W

Widespace Mit Widespace werden sämtliche Zeichen bezeichnet, die einen Leerraum erzeugen. Hierzu gehören Leerzeichen, Tabulatoren oder auch Zeilenumbrüche. (S. 50)

wissensbasiert Als wissensbasiert bezeichnet man Verfahren, die Daten aufgrund eines bestimmten Vorwissens interpretieren. Dieses Vorwissen kann implizit eingebracht werden, z.B. durch Codieren des Wissens in einen Algorithmus, oder explizit, z.B. in Form einer Wissensbasis, die unabhängig vom Datenverarbeitungsverfahren bearbeitet werden kann. Manchmal ist das Wissen auch nur im Operateur vorhanden, der das Verfahren nur in einem bestimmten Kontext einsetzt. In gewissem Sinne ist GEOAIDA solch ein Operateur. (S. 39)

Literaturverzeichnis

- [1] *HALCON*. URL <http://www.mvtec.com/halcon/>.
- [2] *Khoros*. URL <http://www.khoral.com>.
- [3] Achen, M.: *Untersuchungen über Nutzungsmöglichkeiten von Satellitenbilddaten für eine ökologisch orientierte Stadtplanung am Beispiel Heidelberg*. Selbstverlag des Geographischen Institutes der Universität Heidelberg, Heidelberg, 1993.
- [4] Ackermann, F.: *Digital Image Correlation: Performance and potential Application in Photogrammetry*. In *Photogrammetric Record*, Bd. 11, S. 429–439. 1985.
- [5] Aviad, Z., Carnine, P.: *Road Finding for Road-Network Extraction*. In *Computer Vision and Pattern Recognition*, S. 814–819. IEEE, 1988.
- [6] Baatz, M., Heynen, M., Hofmann, P., Lingenfelder, I., Mimler, M., Schäpe, A., Willhauck, G.: *eCognition User Guide*. Definiens AG, München, 2000.
- [7] Baatz, M., Schäpe, A.: *Object-oriented and multi-scale image analysis in semantic networks*. In *Proc. 2nd International Symposium on Operationalisation of Remote Sensing Conference and Exhibition*. ITC, Enschede, 1999.
- [8] Baatz, M., Schäpe, A.: *Multiresolution segmentation – an optimization approach for high quality multi-scale image segmentation*. In Strobl, J., B. T. G. G., Hg., *Angewandte Geographische Informationsverarbeitung –*

- Beiträge zum AGIT-Symposium Salzburg*, S. 12–23. Wichmann, Karlsruhe, 2000.
- [9] Baillard, C. e. a.: *Segmentation of Urban Scenes from Aerial Stereo Imagery*. In *Proc. of 14th Int. Conf. on Pattern Recognition*. Brisbane, Australia, 1998.
- [10] Baltsavias, E. P.: *Hierarchical Multiphoto Matching and D.T.M Generation*. In *Int. Archives for Photogrammetry and Remote Sensing*, Bd. 27 von *Com.III*. Kyoto, 1988.
- [11] Barnsley, M., Barr, S.: *A graph-based structural pattern recognition system to infer land use from fine spatial resolution land cover data*. *Computers, Environment and Urban Systems*, Bd. 21:209–225, 1997.
- [12] Barzohar, M., Cooper, D.: *Automatic Finding of Main Road in Areal Images by Using Geometric - Stochastic Models and Estimation*. In *Computer Vision and Pattern Recognition*, S. 459–464. IEEE, 1993.
- [13] Bauer, T., Steinnocher, K.: *Per-parcel land use classification in urban areas applying a rule-based technique*. *GIS*, Bd. 6:24–27, 2001.
- [14] Baumgartner, A., Eckstein, W., Heipke, C., Hinz, S., Mayer, H., Radig, B., Steger, C., Wiedemann, C.: *TUM-Research on Road Extraction*. Christian Heipke and Helmut Mayer, 1999.
- [15] Baumgartner, A., Steger, C., Mayer, H., Eckstein, W., Ebner, H.: *Automatic Road Extraction based on Multi-Scale, Grouping, and Context*. In *Photogrammetric Engineering & Remote Sensing*, Bd. 65, S. 777–786. 1999.
- [16] Belknap, R., Hanson, A., Riseman, E.: *The information fusion problem and a rule-based hypotheses applied to complex aggregations of image events*. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, S. 22–26. Miami, 1986.
- [17] Beveridge, R., Hanson, A., Riseman, E.: *Segmenting images using localized histograms*. Coins technical report, University of Massachusetts at Amherst, 1986.

-
- [18] Blaschke, T.: *Objectextraction und regelbasierte Klassifikation von Fernerkundungsdaten: Neue Möglichkeiten für GIS-Anwender und Planer*. In Schrenk, M., Hg., *Beiträge zum 5. Symposium zur Rolle der Informationstechnologie in der und für die Raumplanung*, S. 153–162. Wien, 2000.
- [19] Blaschke, T., Lang, S., Lorup, E., Strobl, J., Zeil, P.: *Object-oriented image processing in an integrated GIS/remote sensing environment for environmental applications*. In Cremers, A., Greve, K., Hg., *Umweltinformatik '00*. Metropolis, Bonn, 2000.
- [20] Brenner, C.: *Dreidimensionale Gebäuderekonstruktion aus digitalen Oberflächenmodellen und Grundrissen*. Dissertation, Fakultät für Bauingenieur- und Vermessungswesen, Universität Stuttgart, Stuttgart, 2000.
- [21] Brooks, R. A.: *Model-Based Three-Dimensional Interpretations of Two-Dimensional Images*. In *IEEE Trans. on Pattern Analysis and Machine Intelligence* [80], S. 140–150.
- [22] Burns, J., Hanson, A., Riseman, E.: *Extracting linear features*. In *7th International Conference on Pattern Recognition*, Bd. I, S. 482–485. Montreal, 1984.
- [23] Burns, J., Hanson, A., Riseman, E.: *Extracting straight lines*. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, Bd. 8:425–455, 1986.
- [24] Bückner, J.: *Model Based Road Extraction for the Registration and Interpretation of Remote Sensing Data*. In *ISPRS Commission IV Symposium on 'GIS - Between Visions and Applications'*. Stuttgart, Germany, 7.9.-10.9.98, 1998.
- [25] Bückner, J.: *Ein wissensbasiertes System zur automatischen Extraktion von semantischen Informationen aus digitalen Fernerkundungsdaten*. Dissertation, Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, Universität Hannover, Hannover, 2002.

-
- [26] Bückner, J., Müller, S., Pahl, M., Stahlhut, O.: *Semantic Interpretation of Remote Sensing Data*. In *Proc. ISPRS2002*. Graz, Austria, 2002.
- [27] Bückner, J., Pahl, M., Stahlhut, O.: *Radiometric Equalization of Remote Sensing Data by Utilization of Laserscan Data*. In *CCECE-2001 Canadian Conference on Electrical and Computer Engineering*. Toronto, Canada, 2001.
- [28] Bückner, J., Pahl, M., Stahlhut, O., Liedtke, C.-E.: *GeoAIDA - A Knowledge Based Automatic Image Data Analyser for Remote Sensing Data*. In *CIMA 2001*. Bangor, Wales, UK, 2001.
- [29] Bückner, J., Pahl, M., Stahlhut, O., Liedtke, C.-E.: *A Knowledge-Based System for Context Dependent Evaluation of Remote Sensing Data*. In *Pattern Recognition, 24th DAGM Symposium*, Nr. 2449 in LNCS, S. 58–67. Springer, Zurich, Switzerland, 2002.
- [30] Clement, V., Giraudon, G., Houzelle, S., Sadakly, F.: *Interpretation of Remotely Sensed Images in a Context of Multisensor Fusion Using a Multispecialist Architecture*. *IEEE Trans. on Geoscience and Remote Sensing*, Bd. 31(4):779–791, 1993.
- [31] Coleman, G. B., Andrews, H. C.: *Image segmentation by clustering*. *Proceedings IEEE*, Bd. 67:773–785, 1979.
- [32] Collins, R. T., Jayens, C. O., Cheng, Y.-Q., Wang, X., Stolle, F., Riseman, E. M., Hanson, A. R.: *The Ascender System: Automated Site Modeling from Multiple Aerial Images*. In *Computer Vision and Image Understanding*, Bd. 72, S. 143–162. 1998.
- [33] Curlander, J. C., McDonough, R. N.: *Synthetic Aperture Radar, Systems and Signal Processing*. John Wiley & Sons, New York, 1991.
- [34] Draper, B., Collins, R., Brolio, J., Hanson, A., Riseman, E.: *The Schema System*. *International Journal of Computer Vision*, Bd. 2(3):209–250, 1989.
- [35] ERDAS Inc. - Atlanta,: *ERDAS Field Guide*, 1999.

-
- [36] Fischer, A., Kolbe, T. H., Lang, F., Cremers, A. B., Förstner, W., Plümer, L., Steinhage, V.: *Extracting Buildings from Aerial Images Using Hierarchical Aggregation in 2D and 3D*. In *Computer Vision and Image Understanding*, Bd. 72, S. 185–203. 1998.
- [37] Gerke, M., Heipke, C., Straub, B.-M.: *Building extraction from aerial imagery using a generic scene model and invariant geometric moments*. In *Proceedings IEEE/ISPRS Joint Workshop on Remote Sensing and data fusion over urban areas*, S. 85–89. 2001.
- [38] Gimel'farb, G. L., Zalesny, A. V.: *Probabilistic models of digital region maps based on Markov random fields with short and long-range interaction*. In *Pattern Recognition Letters*, Bd. 14, S. 789–797. 1993.
- [39] Gorte, B.: *Probabilistic Segmentation of Remotely Sensed Images*. Dissertation, ITC, Twente, 1998.
- [40] Gottlob, G., Frühwirth, T., Horn, W.: *Expertensysteme*. Springer Verlag, Wien, 1990.
- [41] Grau, O.: *Wissensbasierte 3D-Analyse von Gebäudeszenen aus mehreren frei gewählten Stereofotos*. Dissertation, Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, Universität Hannover, Hannover, 2000.
- [42] Growe, S.: *Wissensbasierte Interpretation multitemporaler Luftbilder*. Dissertation, Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, Universität Hannover, Hannover, 2000.
- [43] Growe, S., Tönjes, R.: *A Knowledge Based Approach to Automatic Image Registration*. In *Proc. of International Conference on Image Processing (ICIP'97)*, Bd. III. Santa Barbara, USA, 1997.
- [44] Grünreich, D.: *ATKIS - A Topographic Information System as a Basis for a GIS and Digital Cartography in West Germany*. *Geol. Jb.*, Bd. A122:207–215, 1992.

-
- [45] Haala, N., Brenner, C.: *Interpretation of Urban Surface Models Using 2D Building Information*. In *Computer Vision and Image Understanding*, Bd. 72, S. 204–214. 1998.
- [46] Hanson, A., Riseman, E.: *The VISIONS Image-Understanding System*. In Brown, C., Hg., *Advances in Computer Visions*, S. 1–114. Lawrence Erlbaum Associates, Hillsdale, 1988.
- [47] Hanson, A., Riseman, E. M.: *Computer Visions Systems*, Kap. VISIONS: A Computer System for Interpreting Scenes, S. 303–334. Academic Press, New York, 1978.
- [48] Haralick, R. M., Riseman, E. R.: *Image segmentation techniques*. In *Computer Vision, Graphics, and Image Processing*, Bd. 29, S. 100–132. 1985.
- [49] Hartmann, G.: *Recognition of Hierarchically Encoded Images by Technical and Biological Systems*. In *Biological Cybernetics*, Bd. 57, S. 73–84. 1987.
- [50] Healey, G., Binford, T. O.: *The role and use of color in a general vision system*. In *DARPA Image Understanding Workshop*, S. 599–613. Los Angeles, 1987.
- [51] Henricsson, O.: *The Role of Color Attributes and Similarity Grouping in 3-D Building Reconstruction*. In *Computer Vision and Image Understanding*, Bd. 72, S. 163–184. 1998.
- [52] Horowitz, S. L., Pavlidis, T.: *Picture segmentation by a tree traversal algorithm*. *Journal of the ACM*, Bd. 23:368–388, 1976.
- [53] Illert, A.: *Konzepte zur Fortführung des ATKIS*. *Nachrichten aus dem Karten- und Vermessungswesen*, S. 61–74, 1994. Reihe 1, Heft 111.
- [54] Jain, A. K., Dubes, R. C.: *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ, 1988.
- [55] Kettig, R. L., Landgrebe, D. A.: *Computer Classification of Remotely Sensed Multispectral Image Data by Extraction and Classification of Homogeneous Objects*. In *IEEE Transactions on Geoscience Electronics*, Bd. GE-14, S. 19–26. 1976.

-
- [56] Klinker, G. J., Shafer, S. A., Kanade, T.: *A physical approach to color image understanding*. *International Journal of Computer Vision*, Bd. 4:7–38, 1990.
- [57] Kohl, C., Hanson, A., Riseman, E.: *A goal-directed intermediate level executive for image interpretation*. In *10th International Joint Conference on Artificial Intelligence*, Bd. II, S. 811–814. Milan, 1987.
- [58] Kollwe, M., Spitzner, H.: *A Link between Multispectral Remote Sensing, Image Processing and Ecological Analysis of Landscape Elements*. In *Proc. of IGARSS'95*, S. 291–293. IEEE, 1995.
- [59] Kraus, K.: *Photogrammetrie*. Dümmlerbuch, Bonn, 1996.
- [60] Kummert, F.: *Flexible Steuerung eines sprachverstehenden Systems mit homogener Wissensbasis*. Infix Verlag, Sankt Augustin, 1992.
- [61] Kummert, F.: *Interpretation von Bild- und Sprachsignalen – Ein hybrider Ansatz*. Shaker Verlag, 1998.
- [62] Landesvermessungsamt Nordrhein-Westfalen, Bonn: *ATKIS-Gesamtdokumentation, Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV)*, 1989.
- [63] Lang, F.: *Geometrische und semantische Rekonstruktion von Gebäuden durch Ableitung von 3D Gebäudeecken*. Dissertation, Universität Bonn, 1999. Reihe Informatik, Shaker Verlag.
- [64] Le Besnerais, G., Oriot, H.: *Disparity estimation for high resolution stereoscopic reconstruction using the GNC approach*. In *Proc. IEEE Conf. on Image Processing*. Chicago, Illinois, 1998.
- [65] Leberl, F. W.: *Radargrammetric Image Processing*. Artech House, Norwood, MA, 1990.
- [66] Lehrer, N., Reynolds, G., Griffith, J.: *Initial hypothesis formation in image understanding using an automatically generated knowledge base*. In *Proc. DARPA Image Understanding Workshop*, S. 521–537. Morgan Kaufmann Publishers, Inc., 1987.

- [67] Liedtke, C.-E., Bückner, J., Grau, O., Growe, S., Tönjes, R.: *AIDA: A System for the Knowledge Based Interpretation of Remote Sensing Data*. In *3rd Int. Airborne Remote Sensing Conference & Exhibition*, Bd. II, S. 313–320. Copenhagen, Denmark, 1997.
- [68] Liedtke, C.-E., Bückner, J., Pahl, M., Stahlhut, O.: *Knowledge based system for the interpretation of complex scenes*. In *Third International Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images*. Ascona, Schweiz, 2001.
- [69] Liedtke, C.-E., Ender, M.: *Wissensbasierte Bildverarbeitung*. Springer-Verlag, 1989.
- [70] Lütjen, K.: *BPI: Ein Blackboard-basiertes Produktionssystem für die automatische Bildauswertung*. In Hartmann, G., Hg., *Mustererkennung 1986*, 8. DAGM-Symposium, S. 164–168. Springer-Verlag, Berlin, 1986.
- [71] Matsuyama, T.: *Knowledge-Based Aerial Image Understanding Systems and Expert Systems for Image Processing*. In *IEEE Transactions on Geoscience and Remote Sensing*, Bd. GE-25, S. 305–316. 1987.
- [72] Matsuyama, T., Hwang, V. S.-S.: *SIGMA: A Knowledge-Based Aerial Image Understanding System*. Plenum Press, New York/London, 1990.
- [73] McKeown, D., Harvey, W. A., McDermott, J.: *Rule-Based Interpretation of Aerial Imagery*. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, Bd. 7(5):570–585, 1985.
- [74] Meinel, G., Neubert, M., Reder, J.: *Pixelorientierte versus segmentorientierte Klassifikation von IKONOS-Satellitenbilddaten – ein Methodenvergleich*. *Zeitschrift für Photogrammetrie, Fernerkundung und Geoinformationen*, (3), 2001.
- [75] Musser, D. R.: *The STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library*. Longman Publishing Group, 2001. ISBN 0-201-37923-6.

-
- [76] Müller, S., Feitosa, R. Q., Mota, G. L. A., da Silva, V. V., da Costa, D. P., Tanisaki, K.: *GeoAIDA applied for spot satellite image interpretation*. Berlin, Germany. Angenommene Veröffentlichung.
- [77] Münkkel, H.: *Bewertung von Adaptionstrategien bei Bildsegmentierungsverfahren*. Dissertation, Universität Hannover, Stuttgart, 2001.
- [78] Nagin, P., Hanson, A., Riseman, E.: *Studies in global and local histogram-guided relaxations algorithms*. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, Bd. 3:263–277, 1982.
- [79] Nevatia, R.: *A color edge detector and its use in scene segmentation*. *IEEE Transaction on Systems, Man, Cybernetics*, Bd. 7(11):820–826, 1977.
- [80] Niemann, H., Sagerer, G., Schröder, S., Kummert, F.: *ERNEST: A Semantic Network System for Pattern Understanding*. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Bd. 12(9):883–905, 1990.
- [81] Novak, C., Shafer, S.: *Color edge detection*. In *Proc. Image Understanding Workshop*, S. 35–37. 1987.
- [82] Ohlander, R., Price, K., Reddy, D. R.: *Picture segmentation using a recursive region splitting method*. *Computer Graphics and Image Processing*, Bd. 8:313–333, 1978.
- [83] Pakzad, K.: *Wissensbasierte Interpretation von Vegetationsflächen aus multitemporalen Fernerkundungsdaten*. Dissertation, Bayerischen Akademie der Wissenschaften, München, 2001.
- [84] Pietikäinen, M., Harwood, D.: *Segmentation of color images using edge-preserving filters*. In Cappellini, V., Marconi, R., Hg., *Advances in Image Processing and Pattern Recognition*, S. 94–99. North-Holland, 1986.
- [85] Quint, F.: *Kartengestützte Interpretation monokularer Luftbilder*. Dissertation, Universität Karlsruhe, Karlsruhe, 1997.
- [86] Quint, F.: *MOSES: A structural approach to aerial image understanding*. In Gruen, A., Baltsavias, E., Henricsson, O. P., Hg., *Automatic Extraction*

- of Man-Made Objects from Aerial and Space Images (II)*, S. 323–332. Birkhäuser, Basel, 1997.
- [87] Rehrmann, V.: *Stabile, echtzeitfähige Farbbildauswertung*. Verlag Fölbach, Koblenz, 1994.
- [88] Rehrmann, V.: *Segmentierung von Farbbildern*. In *1. Heidelberger Bildverarbeitungsforum*. Universität Koblenz-Landau, Universität Heidelberg, 1995.
- [89] Rehrmann, V., Priese, L.: *Fast and Robust Segmentation of Natural Color Scenes*. In *3rd Asian Conference on Computer Vision*. Hongkong, 1998.
- [90] Reynolds, G., Beveridge, J.: *Searching for geometric structure in images of natural scenes*. In *Proc. DARPA Image Understanding Workshop*, S. 257–271. Morgan Kaufmann Publishers, Inc., 1987.
- [91] Riseman, E., Hanson, A.: *A methodology for the development of knowledge-based vision systems*. In *Proc. of the IEEE Workshop on Principles of Knowledge-Based Systems*. Denver, 1984.
- [92] Riseman, E., Hanson, A., Belknap, R.: *The information fusion problem: Forming token aggregations across multiple representations*. Coins technical report, University of Massachusetts at Amherst, 1987.
- [93] Sagerer, G., Niemann, H.: *Semantic Networks for Understanding Scenes*. In *Advances in Computer Vision and Machine Intelligence*. Plenum Press, New York/London, 1997.
- [94] Schneider, W., Bartl, R., Burger, H., Steinwendner, J., Suppan, F.: *Aerial Photo Interpretation and Satellite Image Analysis in Agricultural Sciences*. *Vermessung & Geoinformation*, Bd. 84(2):164–173, 1996.
- [95] Schreier, G.: *SAR Geocoding: Data and Systems*. Wichmann, Karlsruhe, 1993.
- [96] Schwiedefsky, E., Ackermann, F.: *Photogrammetrie*. B.G.Teubner, Stuttgart, 1976.

-
- [97] Shahrabi, B. A.: *Automatic Recognition and 3D Reconstruction of Buildings from Digital Imagery*. Dissertation, Fakultät für Bauingenieur- und Vermessungswesen, Universität Stuttgart, Stuttgart, 2000.
- [98] Skarbek, W., Koschan, A.: *Color Image Segmentation - A Survey*. Techn. Ber. 94-32, Fachbereich Informatik, TU Berlin, 1994.
- [99] Stevens, W.: *Programmieren von UNIX-Netzwerken*. Hanser Fachbuch, 2000. ISBN 3-446-21334-1.
- [100] Stilla, U., Michaelsen, E.: *Semantic modelling of man-made objects by production nets*. In Gruen, A., Baltsavias, E., Henricsson, O. P., Hg., *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, S. 43–52. Birkhäuser, Basel, 1997.
- [101] Tönjes, R.: *Wissensbasierte Interpretation und 3D-Rekonstruktion von Landschaftsszenen aus Luftbildern*. Dissertation, Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, Universität Hannover, Hannover, 1998.
- [102] Vosselmann, G., de Knecht, J.: *Road Tracing by profile matching and Kalman filtering*. In Grün, A., Kübler, O., Agouris, P., Hg., *Automatic Extraction of Man-Made Objects from Aerial and Space images*, S. 265–274. Birkhäuser, Basel, 1995.
- [103] Weidner, U.: *Digital Surface Models for Building Extraction*. In Grün, A., Hg., *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, S. 193–202. Ascona, 1997.
- [104] Weidner, U., Förstner, W.: *Towards Automatic Building Extraction from High-Resolution Digital Elevation Models*. *PandRS*, Bd. 50(4):38–49, 1995.
- [105] Winston, P. H.: *Artificial Intelligence - 3rd Edition*. Addison-Wesley, Massachusetts, 1992.
- [106] Zucker, S. W.: *Region growing: Childhood and adolescence*. In *Computer Graphics and Image Processing*, Bd. 5, S. 382–399. 1976.

Lebenslauf

Persönliche Daten

Name: Martin Pahl
Geburtsdatum: 14. März 1968
Geburtsort: Oldenburg (Nds)
Staatsangehörigkeit: deutsch
Familienstand: verheiratet, 3 Kinder

Schulausbildung

1974–1978 Grundschule II in Lemgo
1978–1987 Marianne-Weber-Gymnasium in Lemgo
Abschluss: Abitur

Wehrersatzdienst

1987-1989 Zivildienst, Diakoniestation Lemgo

Hochschulausbildung

1989–1995 Studium der Elektrotechnik,
Schwerpunkt Mess- und Regelungstechnik
Abschluss: Diplom (Dipl.-Ing.)

Berufstätigkeit

seit 1995 Wissenschaftlicher Mitarbeiter an der Universität Hannover,
Institut für Theoretische Nachrichtentechnik
und Informationsverarbeitung