Institut für Geodäsie und Geoinformation
Bereich Photogrammetrie

Ein hierarchischer Ansatz zur Interpretation von Gebäudeaufnahmen

Inaugural-Dissertation

zur

Erlangung des Grades

Doktor-Ingenieur

(Dr.-Ing.)

der

Hohen Landwirtschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität

zu Bonn

vorgelegt am 8. August 2011

von

Martin Drauschke

aus Berlin

Referent:	Prof. Dr.-Ing. Dr. h.c. Dr. h.c. Wolfgang Förstner
Korreferent:	Prof. Dr. rer. nat. Lutz Plümer
Tag der mündlichen Prüfung:	28. November 2011
Erscheinungsjahr:	2012
	Diese Dissertation wurde auf dem Hochschulschriften-server der ULB Bonn http://hss.ulb.uni-bonn.de/diss_online elektronisch publiziert.

In Gedenken an meine Oma
Für Andreas, meine Mutter und meinen Neffen Moritz

Danksagung

Ich möchte mich bei meinem Doktorvater, Prof. Dr.-Ing. Wolfgang Förstner, für die Betreuung dieser Arbeit bedanken. Er hat mir wesentlich bei der Themenfindung geholfen und mich bei der wissenschaftlichen Formulierung des Konzepts und bei dessen Umsetzung mit zahlreichen Anregungen und lehrreichen Kommentaren unterstützt. Das Arbeitsklima an seinem Lehrstuhl empfinde ich motivierend und der wissenschaftlichen Neugier sehr förderlich. Vielen Dank dafür, dass ich diesem Kollegium für einige Jahre angehören durfte. Allen meinen Kollegen sei an dieser Stelle ebenfalls gedankt für die nette Atmosphäre am Institut. Insbesondere der freundschaftliche Umgang untereinander hat mir während meiner Tätigkeit am Bonner Institut sehr gut getan. Gern denke ich an die lebhaften Diskussionen zurück, die ich vor allem mit meinen Projektpartnern Filip Korč und Susanne Wenzel sowie mit allen anderen Kolleginnen und Kollegen geführt habe, insbesondere mit Thomas Läbe, Ribana Roscher, Hanns-Florian Schuster und Michael Ying Yang. Ebenso möchte ich mich bei Barbara Förstner, Heidi Hollander und Monika Tüttenberg für ihre aufmunternden Worte herzlich bedanken.

Bei Prof. Dr. Lutz Plümer möchte ich mich für die Übernahme des Zweitgutachtens bedanken. Durch die gemeinsamen Projekttreffen war er schon frühzeitig mit dem Thema dieser Arbeit vertraut und hat durch kritische Nachfragen einiges Nachdenken meinerseits gefördert. Auch die anderen Projektpartner, ob im Skalen-Bündel oder bei eTRIMS, haben sicher Spuren bei mir und meiner Arbeit hinterlassen.

Einen Großteil dieser Arbeit habe ich neben meiner Tätigkeit an der Universität der Bundeswehr geschrieben. Ich möchte mich deshalb auch bei Prof. Dr.-Ing. Helmut Mayer und meinen dortigen Kollegen bedanken, dass sie mir ab und zu den notwendigen Freiraum eingeräumt haben. Zudem habe ich hier einen zusätzlichen Blickwinkel auf manche Forschungsthemen erhalten, was diese Arbeit gewiss an der einen oder anderen Stelle verbessert hat.

Die Themenstellung bzgl. der Interpretation von Gebäudebildern hat auch zur Intensivierung von zwei Freundschaften geführt: Bei Martin Bredenbeck habe ich mich über kunstgeschichtliche Aspekte meines Themas informiert und bei Matthias Gauggel über Architektur. Ebenso haben die Gespräche mit meinen Freunden und ehemaligen Kommilitonen Bernhard Fisseni, Meike Jungebloed und Kai Oliver Heuer dazu beigetragen, den Einsatz von statistischen Methoden in anderen Anwendungsgebieten zu diskutieren. Andreas Of hat diese Arbeit auf sprachliche Korrektheit überprüft und mich dabei auf einige unklare Formulierungen hingewiesen. Vielen Dank auch für diese hilfreichen Bemerkungen.

Abschließend möchte ich mich noch bei Andreas Of und meiner Mutter bedanken, die mir in den vergangenen Jahren regelmäßig Mut zugesprochen und dadurch Kraft gegeben haben, diese Arbeit abzuschließen. Ich werde bald wieder mehr Zeit für euch und meine anderen Freunde und Verwandten haben.

Zusammenfassung

Durch die Fortschritte bei der 3D-Visualisierung von Landschaften und urbanen Räumen in den vergangenen Jahren wächst die Nachfrage nach semantischen und detailgenauen 3D-Stadtmodellen. Die Interpretation von Digitalbildern ist dabei ein wesentlicher Schritt zur Informationsgewinnung für die Verfeinerung und Aktualisierung der Gebäudemodelle. Wegen der großen Datenmenge muss diese Bildinterpretation soweit möglich automatisiert durchgeführt werden, um kostengünstig und zeitnah Ergebnisse liefern zu können.

Bei der Interpretation eines Gebäudebildes wird das gegebene Farbbild in eine Klassifikationskarte überführt, in der die Bildpixel entsprechend ihrer Semantik eingefärbt werden. Dieser Vorgang ist wegen der fehlenden 3D-Information über das Gebäude und seine Umgebung sowie der starken Ähnlichkeiten zwischen einigen Gebäudeteilen besonders schwierig. Hinzu kommt noch die große Vielfalt der Gebäude in Bezug auf deren Größe, Beschaffenheit und Stil sowie in der Art und Anzahl der Fassadenbestandteile und Dachaufbauten und deren räumlicher Anordnung.

Das in der vorliegenden Arbeit vorgestellte Verfahren untersucht die Leistungsfähigkeit der Erkennung von Objekten und Objektteilen in Gebäudeaufnahmen bei Verwendung der Bestandteilshierarchien. Diese werden in eine Hierarchie von segmentierten Bildregionen abgebildet und zur Konstruktion eines Bedingten Bayes Netzes verwendet. Die Gesamtwahrscheinlichkeit dieses Bedingten Bayes-Netzes wird effizient als Produkt von zwei Termen bestimmt, die einerseits durch die Klassifikation der Regionen auf der Basis regionenspezifischer Merkmale und andererseits aus den Klassenzugehörigkeiten von in der Regionenhierarchie benachbarten Regionen bestimmt werden. Für diesen Ansatz werden folgende Teilschritte zusammengeführt: die Segmentierung von geometrisch präzisen Regionen in mehreren Maßstabsebenen des Bildes und deren hierarchische Anordnung, die Auswahl stabiler Regionen, der Vergleich von Regionen mit manuell angefertigten Annotationen, die Extraktion und Selektion geeigneter Merkmale sowie die Klassifikation der Regionen.

Die stabilen Regionen der hierarchischen Bildsegmentierung eignen sich gut zur Detektion von komplexen Objekten und ihren Bestandteilen und bilden die Relationen der Bestandteilshierarchie von Objekten in die Regionenhierarchie ab. Die klassenspezifischen Verteilungen der regionenspezifischen Merkmale unterscheiden sich und eignen sich für eine maßstabsabhängige Klassifikation der Regionen. Dazu wird die Methode der Alternierenden Entscheidungsbäume vom Zwei- auf den Mehrklassenfall verallgemeinert. Diese Klassifikationsergebnisse werden als initiale Belegung der Zufallsvariablen in das Bedingte Bayes-Netz integriert. Für die anschließende Inferenz der Information kann wegen der baumartigen Struktur des Bayes-Netzes ein sehr einfacher Algorithmus verwendet werden, der lediglich einen zweimaligen Durchlauf durch den Baum vorsieht.

Die Bildinterpretation nach Anwendung des Bayes-Netzes liefert leicht verbesserte Ergebnisse für die Klassifikation der Regionen. Das Verfahren wurde auf insgesamt vier Datensätzen mit zusammen über 600 Bildern evaluiert. Bei der Segmentierung und Konstruktion der Regionenhierarchie werden sehr zufrieden stellende Ergebnisse erzielt, da zu 70 bis 90% der vorhandenen Objekte passende Regionen und Relationen zwischen den Regionen gefunden werden können. Auf dem Benchmark-Datensatz von Korč & Förstner (2009) erzielen wir bei der Klassifikation ohne Nutzung der Bestandteilshierarchie mit sieben Klassen einen akzeptablen Erfolg von 54.9%. Allerdings werden die drei besonders häufig vorkommenden Klassen Gebäude, Fenster und Vegetation extrem durch den Klassifikator bevorzugt. Durch die Propagierung der Information im Bedingten Bayes-Netz wird die Erfolgsrate auf 60.7% verbessert, aber die starken Unterschiede bei den Fehlklassifikationen bleiben bestehen.

Summary

The recent progress in 3D visualization of landscapes and urban spaces has increased the demand for very detailed 3D city models with semantics. Therefore, the interpretation of digital imagery is an important step for gaining information used to enrich building models or for updating them. Due to the large amount of data, image interpretation should be done automatically as far as possible for a fast and affordable delivery of results.

Interpreting building images means to derive a symbolic map for a given color image where each pixel is visualized by its semantic. This procedure is extremely difficult if no additional 3D information about the objects in the scene is used. Regarding buildings, their parts often appear too similar, especially w. r. t. their shape or the color of the representing pixels. Furthermore, the large variability among buildings w. r. t. their size, nature and style as well as the kind and number of facade parts and roof structures and their spatial arrangement.

In the approach of this work, the capability of detecting objects and their parts in building images is analyzed with a focus on the integration of object hierarchies. The relations between the objects and their parts are mapped into a hierarchy of segmented image regions which is further used to construct a Bayesian network. Features of these regions are determined, thus the Bayesian network can be formulated as a conditional one. The overall probability of this Conditional Bayesian Network can be efficiently determined as product of two terms. The first term is derived from classification results based on observed region-specific features, the second term is derived from the co-occurrence of class targets defined by neighbored regions in the region hierarchy. The following methods are combined within this approach: segmentation of geometrically precise regions at several image scales and their hierarchical arrangement, choosing stable regions for further processes, comparison between automatically segmented image regions with manually labeled annotations, extraction and selection of appropriate features and classification of the regions.

The stable regions of the hierarchical image segmentation are suitable for detecting complex objects and their parts. The relations of the object hierarchy are mostly mapped into the region hierarchy. The class-specific distributions of the region-specific features can be distinguished from each other and, therefore, can be used for scale-dependent classification of the stable regions. This is done by applying alternating decision trees, which have been extended from binary to multiclass classification. The results are used for initializing the states of the random variables of the Conditional Bayesian Network. A simple algorithm can be applied for propagating the information through the net because of its tree-like structure.

The image interpretation yields slightly better results for region classification after applying the Bayesian network. The approach has been tested and evaluated on four data sets with a total of 600 images. Obtained results on image segmentation and region hierarchy construction are satisfying since 70 to 90% of all objects of interest are represented by at least one of the segmented regions, and the relations between the regions reflect the object hierarchy. On the benchmark dataset of Korč & Förstner (2009), the overall classification result of 54.9% is obtained when classifying the regions without considering the hierarchical relations and belonging to seven classes. But the three classes vegetation, building and window which appear in images more often than other classes are favored by the classifier. After propagating the information through the Conditional Bayesian Network the classification rate is increased to 60.7% but the obvious differences regarding classification mistakes remain.

Inhaltsverzeichnis

1. Einführung

1.1. Motivation und Ziel der Arbeit

Die automatische Interpretation von digitalen Gebäudebildern ist eine komplexe Aufgabe, die seit ein paar Jahren eine hohe Aufmerksamkeit in der photogrammetrischen Forschung und im Computer Vision hat. Die Schwierigkeit liegt einerseits an der Vielfalt der Gebäude, die von Bungalows über Reihen- und Mehrfamilienhäuser bis hin zu Fabriken, Schlössern und Kirchen reicht. Andererseits unterscheiden sich Gebäude auch in der Art und Anzahl der Fassadenbestandteile und Dachaufbauten, deren Größe und Form, deren räumlichen Anordnung und deren Beschaffenheit.

Die starke Beachtung erhält die automatische Interpretation von Gebäudebildern vor allem durch die Erstellung, Aktualisierung und Veredlung von 3D-Stadtmodellen, in denen bis zu mehrere hunderttausend Gebäude modelliert werden. Die 3D-Modellierung urbaner Räume von z. B. Müller *et al.* (2006) oder Wahl *et al.* (2008) ist vor allem auf eine effiziente Visualisierung ausgerichtet, so dass die Modelle vor allem zur Betrachtung des Stadtraums, d. h. bei der Stadtplanung und Verkehrssimulation, in der Tourismusbranche oder von der Filmindustrie verwendet werden können. Bei diesen Anwendungen ist es ausreichend, wenn die entsprechenden Bereiche aus den Gebäudebildern als Textur an die schlichten, rekonstruierten 3D-Modelle projiziert werden. Wenn man zudem die 3D-Stadtmodelle mit Semantik anreichert, dann werden sie auch für andere Wirtschaftsbereiche eine wertschöpfende Technologie: Für die Gebäudeüberwachung, den Rettungsdienst und Katastrophenschutz sowie die Navigation von Fußgängern in Innenstädten können 3D-Stadtmodelle wertvolle Informationen anbieten, wenn sie z. B. Gebäudezugänge oder begehbare Außenbereiche wie Terassen und Balkone aufzeigen. Für die Erstellung von Solaratlanten ist die Erkennung auch sehr kleiner Dachaufbauten wichtig. Reznik (2009), Hohmann *et al.* (2009) und Becker (2010) rekonstruieren Fassadenelemente wie Fenster, Gauben und Ornamente detailliert, was die Visualisierung der Gebäudemodelle erheblich verbessert. In diesen Fällen ist eine Objekterkennung hilfreich für die Wahl und Parametriesierung der Objektmodelle.

Eine weitere Anwendung für die automatische Interpretation von Gebäudebildern sind digitale Karten und Stadtansichten, wie sie *Google Earth* bzw. *Google Street View* anbieten. Bilder werden in diesen Technologien so zusammengesetzt, dass bei Betrachtung echte 3D-Welten suggeriert werden. Analog zu den Stadtmodellen gewinnen auch diese digitalen Ansichten einen Mehrwert, wenn sie mit semantischer Information ergänzt werden.

Die Akzeptanz dieser Technologien ist nur dann hoch, wenn die darin gespeichert Daten korrekt und aktuell sind. Dies setzt fortlaufende Datenerfassung und Datenüberprüfung und damit verbundende Bildauswertung voraus, die aufgrund der Datenfülle automatisch erfolgen sollte.

In dieser Arbeit werden wir die Interpretation von einzelnen Farbbildern explorieren, d. h. wir werden ein RGB-Bild in eine digitale Klassifikationskarte überführen, in der die Bildpixel nach ihrer Semantik eingefärbt werden. Die Zuweisung von mehreren Klassen an ein Pixel ist dabei prinzipiell denkbar, schließlich kann man ein Pixel vom Dach auch als Gebäude inter-

pretieren. Wenn man bei der Klassifikation auch eine Restklasse für nicht weiter definierte Objekte berücksicht, so umfasst diese Definition von Bildinterpretation auch Objektdetektoren, die das Bild in zwei Klassen einteilen: Objektklasse und Restklasse. Da Gebäudebilder viele Objekte zeigen, z. B. das Gebäude mit seinen Teilen Dach und Fassade, Fenster und Türen, aber auch Vegetation, Straße und Himmel, sind wir aber vor allem an einer Bildbeschreibung mit mehreren Klassen interessiert.

Mit der Beschränkung auf die Analyse eines einzelnen Bildes verzichten wir auf die Berücksichtigung zusätzlicher Datenquellen, wie Stereo-Bildpaare, Bildsequenzen und LiDAR-Punktwolken oder Informationen über Gebäudegrundrisse, -höhe oder Nutzungsform, wie sie z. B. in Katasterkarten aufgezeichnet werden. Dass mit diesen zusätzlichen Daten vielversprechende Ergebnisse für die 3D-Rekonstruktion und Interpretation von Gebäudeszenen erzielt werden können, belegen die Arbeiten u. a. von Dick (2001), Ripperda (2009), Becker (2010) und Kluckner (2011).

Moderne Verfahren zur 3D-Rekonstruktion urbaner Räume verwenden heterogenes Bildmaterial. Bartelsen & Mayer (2010) nutzen Mikrokopter zur Datenerfassung und erhalten somit Gebäudeaufnahmen aus drei verschiedenen Blickwinkeln. Es müssen dadurch Fassadenbilder aus terrestrischer Sicht, Schrägbilder aus Traufrandhöhe des Gebäudes und Senkrechtaufnahmen aus der Vogelperspektive interpretiert werden. In Untersuchungen von Agarwal *et al.* (2010) und Frahm *et al.* (2010) werden ebenso beliebige Bilder verwendet, die aus dem Internet heruntergeladen werden. Wir haben uns entschieden, unseren Ansatz für die Bildinterpretation möglichst allgemein zu gestalten, um mit Bildern aus möglichst verschiedenen Blickwinkeln arbeiten zu können. Daher evaluieren wir unser Verfahren sowohl auf terrestrischen Aufnahmen als auch auf Luftbildausschnitten.

Ziel dieser Arbeit ist es, die Interpretation von einzelnen farbigen Gebäudeaufnahmen zu explorieren. In Abb. 1.1 zeigen wir ein Fassadenfoto aus Arlesheim bei Basel in der Schweiz und eine manuell erstellte Interpretationen dieses Bildes. Sie basiert auf der veröffentlichten Bildbeschreibung von Korč & Förstner (2009), berücksichtigt aber nur vier Klassen. Wir zeigen die Gebäude in rot, deren Fenster in blau und die Bäume in hellem grün. Alle übrigen Objekte haben wir in einer Restklasse zusammenfasst und gelb eingefärbt. Ziel dieser Arbeit ist es, eine Bildinterpretation automatisch abzuleiten, die dieser manuellen Interpretation ähnlich ist.

Auch wenn die Interpretation von Bildern einem Menschen bei Fotografien mit guten Aufnahmebedingungen relativ leicht fällt, ist sie aber für den Computer eine sehr herausfordernde Aufgabe, insbesondere beim Erkennen von komplexen Objekten und ihren Bestandteilen. Treisman (1986) hat in ihren Experimenten zur menschlichen Wahrnehmung herausgefunden, dass die Gruppierung von Bildstrukturen eine starke Rolle bei der Erkennung von Objekten in Bildern spielt. Aus diesem Grund basiert unsere Bildinterpretation auf der Analyse homogener Bildregionen, d. h. unser erster Schritt der Bildinterpretation beinhaltet eine zuverlässige Einteilung des Bildes in Regionen.

Ein weiterer wichtiger Bestandteil bei der menschlichen Wahrnehmung komplexer Objekte ist das Erkennen von Bestandteilen und die Analyse von deren räumlicher Anordnung (Biedermann, 1987). Wir werden eine hierarchische Bildsegmentierung durchführen, um so die Aggregationsschritte der Gebäudeteile wie Fenster und Wand zu komplexen Fassaden zu rekonstruieren. Die Umgebung einer segmentierten Region werden wir bei der Interpretation berücksichtigen, wenn wir Merkmale der Region bestimmen, die wir bei der Klassifikation verwenden. Dadurch können wir auf Nachbarschaftsrelationen zwischen den Regionen verzichten, die bei Markoff-Zufallsfeldern starke Berücksichtigung finden, aber die Komplexität

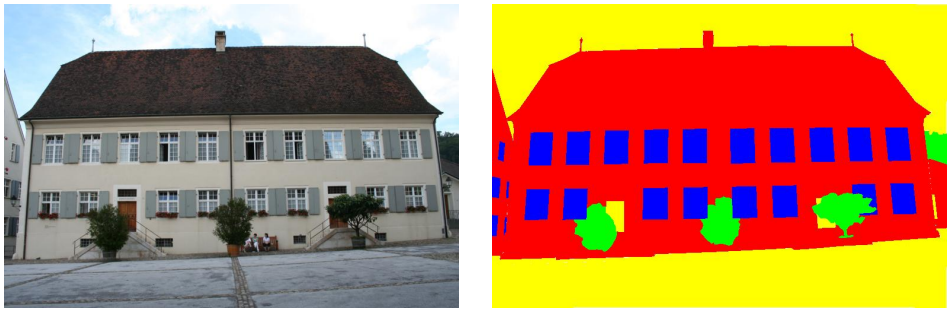


Abbildung 1.1.: Bildinterpretation: Links zeigen wir ein Fassadenfoto aus dem Umland von Basel. Rechts daneben zeigen wir eine manuelle Bildinterpretation dieses Fotos. Darin zeigen wir vier Klassen, u. z. die Gebäude in rot, deren Fenster in blau und die Bäume in hellem grün. Alle übrigen Objekte, d. h. die Türen, den Himmel und die Straße haben wir in einer Restklasse zusammenfasst und gelb eingefärbt.

der Berechnungen signifikant erhöhen, wie Yang *et al.* (2010) erfahren mussten.

Mit der hierarchischen Struktur von Bildregionen modellieren wir ein Bayes-Netz. Analog zu den Bedingten Zufallsfeldern formulieren wir unser Bayes-Netz ebenfalls in Abhängigkeit von beobachteten Merkmalen als Bedingtes Bayes-Netz. Das ermöglicht uns eine saubere statistische Modellierung der Übergangswahrscheinlichkeiten im Bayes-Netz und eine effiziente Berechnung. Unsere Bildinterpretation basiert somit auf zwei Auswerteschritten: Einerseits klassifizieren wir die segmentierten Bildregionen anhand ihrer Merkmale, andererseits wird die Klassifikation unter Berücksichtigung der hierarchischen Struktur der Regionen verbessert. Der erste Anteil entspricht dem Einfluss der unären Potentialfunktionen in Markoff Zufallsfeldern, der zweite dem Einfluss der binären Potentialfunktionen.

Da wir sowohl terrestrische Aufnahmen von Gebäuden als auch Luftbildausschnitte interpretieren wollen, wählen wir einen datengetriebenen Ansatz für die hierarchische Segmentierung und damit für die Struktur unseres Bedingten Bayes-Netzes. Das bringt einige Probleme mit sich: Auch wenn viele der Regionen vollständige Gebäude und vollständige Gebäudeteile zeigen, wird es auch immer wieder Regionen geben, die nur einen Teil eines Objekts zeigen oder mehrere bzw. Teile von mehreren. Zudem können wir nicht garantieren, dass wir die Bestandteilshierarchie von Gebäuden immer fehlerfrei durch die hierarchische Struktur der segmentierten Regionen rekonstruieren.

1.2. Alternative Ansätze

Bevor wir die eigenen Arbeitshypothesen formulieren, geben wir einen kurzen Überblick über den aktuellen Forschungsstand. Wir beginnen mit einigen Arbeiten, in denen Gebäude aus mehreren Bildern rekonstruiert werden. Bei diesen Arbeiten werden in erster Linie 3D-Strukturen interpretiert und mittels vorliegender Bilder verifiziert. Diese Arbeiten zeigen, dass bereits die semantische 3D-Modellierung von Gebäuden wegen der vielen Mehrdeutigkeiten und der großen Vielfalt der Objekte sehr in 3D sehr schwierig ist. Anschließend gehen wir auf aktuelle Forschungsarbeiten in Bezug auf die Bildinterpretation ein, in denen terrestrische Aufnahmen oder Luftbildausschnitte von Gebäuden interpretiert werden. Alternative Ansätze aus der Computer Vision, die bislang nicht oder nur im geringen Umfang an Gebäudebildern

getestet wurden, stellen wir zum Schluss vor.

Interpretation von 3D-Strukturen

Fischer *et al.* (1998) rekonstruieren Gebäude aus mehreren Luftbildern. In einzelnen Luftbildern werden zunächst elementare Bildmerkmale, d. h. Punkte, Linien und Regionen extrahiert. Aggregate von Bildmerkmalen, in denen benachbarte Flächen, Linien und Punkte zu charakteristischen Ecken und Flächen zusammengesetzt werden, werden in den 2D-Bildern bestimmt, deren geometrische Erscheinung in 3D danach aus mehreren Bildern abgeleitet wird. Im 3D-Modell werden diese Aggregate zu größeren Komponenten verbunden, die als Gebäudeteile aufgrund ihrer Dachstruktur erkannt werden. Die Gebäudeteile werden zum Schluss zu Gebäuden zusammengesetzt. Dieses Verfahren weist eine explizite hierarchische Modellierung auf, in der in jedem Schritt die Umgebung der bislang extrahierten Komponenten ausgewertet wird. Die Hypothesen für die Gebäudemodelle werden von Fischer (2005) durch Vergleich mit den Bildstrukturen verifiziert.

Auch Kulschewski (1999) verwendet für seine Gebäudeerkennung Bilder und daraus extrahierte 3D-Information. Er modellierte die Objekterkennung mit einem Bayes-Netz, in dem die 3D-Objekte in 3D-Objektteile zerlegt werden, die als Aspektteile in den Bildern sichtbar sind und erkannt wurden. In den Bildern machen sich diese als Arrangements von Flächen, umrandet von Kanten und Dreibein-Strukturen, die wiederum aus Kanten und Punkten zusammengesetzt werden.

In der systematischen Zerlegung bzw. in umgekehrter Reihenfolge der Rekonstruktion von Objekten aus beobachteten 2D-Merkmalen ähneln sich die beiden Verfahren von Fischer *et al.* (1998) und Kulschewski (1999) stark. Sie unterscheiden sich vielmehr in der Inferenz der Information von 2D-Merkmalen bis hin zur Gebäudeerkennung. Beide Ansätze stoßen an ihre Grenzen, wenn die Gebäudestruktur zu komplex wird, d. h. bei stark strukturierten Gebäudekomplexen mit ineinander übergehenden Dachstrukturen. Außerdem sind sie nur für die Gebäudeextraktion aus Luftbildern geeignet. Eine Extraktion von Fassaden mit Balkonen und Erkern aus terrestrischen Aufnahmen ist mit diesen Algorithmen nicht möglich.

Dick (2001) extrahiert aus mehreren Bildern 3D-Gebäudemodelle, in denen er auch Gebäudeteile wie Fenster und Türen, aber auch für bestimmte Baustile typische Strukturen wie Säulen und Giebel erkennt. Dazu verwendet Dick (2001) neben der Bildinformation auch die 3D-Struktur und das Wissen über Form und Größe der Gebäudeteile. Damit ist seine Modellierung sehr speziell und kann nur in speziellen Szenarien eingesetzt werden.

In Bezug auf die Berücksichtigung von baustiltypischen Elementen und Rekonstruktion von Gebäudefassaden inklusive detailreicher Ornamente ist die Arbeit von Hohmann *et al.* (2009) eine gute Fortsetzung des Ansatzes von Dick (2001). Hohmann *et al.* (2009) stellen einen automatisierbaren Arbeitsablauf vor, in dem sie terrestrische Bilder mit hoher Überlappung und LIDAR-Daten integrieren, um die österreichische Stadt Graz detailgenau zu rekonstruieren. Durch Integration von formalen Grammatiken erzielen sie präzise Beschreibungen für die Gebäudefassaden. Momentan sind noch nicht alle Abläufe des Verfahrens automatisiert, so dass Hohmann *et al.* (2009) immer wieder auf manuelle Interaktion zurückgreifen.

Interpretation von Gebäudeaufnahmen

Jahangiri (2010) schlägt ein vollständig datengetriebenes Verfahren zur Bildinterpretation vor, das aus zwei Stufen besteht. Zuerst findet ein Blob-Detektor kleine bis mittlere Bildregionen konvexer Form, die sich stark von ihrer Umgebung unterscheiden. Dabei greift Jahangiri (2010)

auf die Arbeit von Itti *et al.* (1998) aus der Aufmerksamkeitsforschung zurück: Der Blob-Detektor findet vor allem die Bildregionen, die das menschliche Auge bei der Betrachtung ansteuert. Im zweiten Schritt werden die Ergebnis-Blobs aus dem Bild entfernt und die Lücken im Bild durch ein Inpainting-Verfahren wieder gefüllt. Mittels Mean-Shift wird das Bild nun in große Bildregionen eingeteilt. Die Experimente haben gezeigt, dass der Blob-Detektor bei Fassadenbildern vor allem Fenster und Türen findet und die anschließende Bildsegmentierung das Bild in Fassade, Dach, Vegetation und Himmel zerlegt.

In unseren Augen ist der Ansatz von Jahangiri (2010) sehr gelungen: Er ist universell formuliert, effizient und liefert vielversprechende Ergebnisse. Defizite sehen wir in zwei Bereichen. Erstens verwendet das Verfahren eine relativ starke Glättung des Bildes, so dass kleine und kontrastarme Strukturen unerkannt bleiben: Balkone, Treppen und Gauben bleiben genauso unerkannt wie Fensterscheiben. Daher halten wir auch eine Anwendung bei der Interpration von Luftbildern für ausgeschlossen. Das zweite Defizit des Verfahrens besteht in der fehlenden Präzision für die geometrische Beschreibung der detektierten Objekte. Ein Blob kann eine deutlich größere Fläche einnehmen als das eigentliche Objekt. So kommt es häufig zum Zusammenschmelzen der Blobs von dicht beieinanderliegenden detektierten Objekten.

Das von Jahangiri (2010) entwickelte Verfahren wurde bereits an Bildern getestet, mit denen 3D-Gebäudemodelle texturiert werden. Xu *et al.* (2010) konnten unter Berücksichtigung der nachbarschaftlichen Beziehungen im Bild zwischen den detektierten Blobs viele der Gebäudeteile richtig identifizieren.

Die Interpretation von Fassadenbildern realisiert Korč & Förstner (2008) auf Blockbasis, d. h. die Bildpixel werden zu gleichgroßen, rechteckigen Blöcken zusammengesetzt. Die Klassifikation der Blöcke in Fassade, Himmel und Vegetation wird mit einem bedingten Zufallsfeld durchgeführt, in dem die Potentialfunktionen zur Auswertung der Merkmale der einzelnen Blöcke und der Nachbarschaftsbeziehungen modelliert werden. Zur Grobeinteilung eines Bildes liefert das Verfahren sehr gute Ergebnisse, eine Adaption auf kleinere Blockgrößen und eine Klassifikation von Fassadenteilen wie Fenstern und Türen wurde bislang noch nicht publiziert. Eine Anwendung auf Luftbilder mit einer Klassifikation der Pixelblöcke in Gebäude, Straße und Vegetation halten wir prinzipiell für möglich.

Eine ähnliche Grobeinteilung in Gebäude, Boden, Vegetation und Himmel bei der Auswertung der Fassadenbildern auf Pixelebene haben auch Drauschke & Mayer (2010) und Kluckner (2011) umgesetzt. Der Einfluss der Nachbarschaftbeziehungen wurde bei diesen Verfahren bei der Merkmalsextraktion berücksichtigt, so dass mit diesen Verfahren der starke Anteil der unären Potentialfunktionen an der Klassifikation mit Markoff-Zufallsfeldern demonstriert wird. Experimente zur Interpretation von Luftbildern wurden von Kluckner (2011) bereits durchgeführt. Deren gute Ergebnisse sind aber vor allem auf die Integration mehrer Bilder und die Ableitung von 3D-Information zurückzuführen.

Berg *et al.* (2007) gliedern ihre Bildinterpretation von terrestrischen Stadtaufnahmen in zwei Schritte. An die Bildeinteilung in Grundklassen wie Himmel, Gebäude und Vegetation schließt sich eine verfeinerte Klassifikation von Gebäudeteilen an. Im ersten Schritt werden visuelle Merkmale für die pixelweise Klassifikation verwendet, d. h. der erste Schritt ist datengetrieben modelliert. Anschließend wird Wissen über die Objekte der Szene integriert: Gebäude grenzen sich zu ihrer Umgebung mit klaren Kanten ab. So können horizontal verlaufende Kanten zwischen Himmel und Gebäude zur genaueren Abgrenzung von Instanzen der beiden Objektklassen gesucht werden, vertikale Kanten grenzen das Gebäude an seiner Seite ab. In einer iterativen Vorgehensweise werden nun Modelle zur Beschreibung der in der Szene vorhandenen Klassen gelernt, so dass sich die Bildinterpretation schrittweise verbes-

sert. Abschließend werden die weiteren Bestandteile des Gebäudes detektiert, wobei hier ein Segmetierungsverfahren für die Detektion flächenhafter Teile, eine Kantendetektion für die Begrenzung der Objekte und Ähnlichkeitsanalysen für die Detektion wiederholter Strukturen eingesetzt werden. Die veröffentlichten Ergebnisse sind überzeugend, allerdings geben die Autoren keine Auskunft über die Laufzeiten ihrer Techniken. Eine Adaption auf die Interpretation halten wir grundlegend für denkbar, auch wenn einzelne Schritte wie die Abgrenzung von Himmel und Gebäude durch andere Schritte ersetzt werden müssten.

Das Verfahren von Ripperda (2009) verwendet eine formale Grammatik und ein zufallsbasiertes Reversible Jump Markov Chain Monte Carlo Verfahren (rjMCMC) zum Finden der besten Ableitungsregel für die Rekonstruktion von Fassadenstrukturen. Eigentlich hat Ripperda (2009) das Verfahren für Intensitäts- und Entfernungsbilder konzipiert, wo es auch sehr erfolgreich ist. In einem Experiment auf einzelnen Bildern haben Ripperda & Brenner (2009) aber gezeigt, dass ihr Verfahren ebenfalls akzeptable Ergebnisse erzielen kann. Die Idee zur Verwendung von Grammatiken zur Analyse einer Fassade wurde auch von Reznik (2009), Čech & Šára (2009) und Teboul *et al.* (2010) aufgegriffen, wobei die ersten beiden ihre Grammatiken jeweils nur zur Beschreibung detektierter Fenster in einer Fassade einsetzen. Teboul *et al.* (2010) nutzt die Grammatik neben der Ableitung von Anordnungen der Teile auch zur Integration von Formparametern, die an die Symbole der Grammatik gekoppelt sind.

Wir sehen in den Grammatik-basierten Verfahren zwei Probleme. Einerseits benötigen sie entzerrte Fassadenbilder, in denen die rechteckigen Fassaden ein rechteckiges Abbild haben. So lassen sich die Ableitungsregeln der Grammatiken durch ausschließlich horizontale und vertikale Aggregationsschritte repräsentieren. Andererseits sind Fassaden derart komplex strukturiert, so dass häufig viele verschiedene Ableitungsschritte möglich erscheinen. Somit ist die Komplexität dieser Verfahren relativ hoch. Des Weiteren sehen wir keine Möglichkeit zur Adaption der vorgeschlagenen Grammatiken auf die Analyse von Dachstrukturen in Luftbildern.

Ein einzelnes Luftbild wird eher selten interpretiert, da bei Verwendung zusätzlicher Daten i. d. R. bessere Ergebnisse erzielt werden. Durch die spezielle Aufnahmekonfiguration kann man effizient 3D-Oberflächenmodelle generieren, so dass bei der Datenanalyse in der Regel mehrere Bilder und 3D-Information vorliegen, siehe dazu auch die bereits vorgestellten Arbeiten von Fischer (2005) und Kluckner (2011).

Schuster (2005) segmentiert das Luftbild mit dem Segmentierungsverfahren von Förstner (1994) und analysiert dann Cliquen von segmentierten Regionen. Die sukzessive Aggregatbildung ermöglicht es, sowohl kleine Dachaufbauten zu erfassen als auch größere Dachstrukturen. Die Klassifikation erfolgt durch ein Bayes-Netz, bei dem die bedingten Wahrscheinlichkeiten für das Vorkommen bestimmter Cliquen und Regionen in Abhängigkeit von beobachteten Merkmalen bestimmt werden müssen. Als Merkmale werden auch Charakteristika benachbarter Regionen verwendet, die durch den Nachbarschaftsgraph nach Fuchs (1998) berechnet werden. Das Verfahren hat sein Defizit vor allem in der Wahl der zu Grunde liegenden Segmentierung: Förstner (1994) verwendet ein Homogenitätsmaß zur Segmentierung von Regionen, weshalb in stark texturierten Bereichen keine Regionen segmentiert werden. In anderen (nichtveröffentlichten) Arbeiten hat Schuster sehr gute Klassifikationsergebnisse erzielt, wobei er dazu Bildinformation im infraroten Spektralbereich ausgenutzt hat: den normalisierten differenzierten Vegetationsindex (NDVI) zum Erkennen von Vegetation.

Die Bildinterpretation bei Meixner & Leberl (2010) bezieht sich auf die Fensterdetektion nach Lee & Nevatia (2004) im Anschluss an die erfolgte Fassadenextraktion aus rekonstruierten 3D-Daten. Das Verfahren zur Fensterdetektion basiert auf Auswertung der Gradienten

und findet nur Fenster, die in einem regelmäßigen Gitter angeordnet sind. So sind die guten Detektionsraten in Luftbildern von Innenstädten mit mehrgeschosssigen Gebäuden nachvollziehbar.

Weitere Ansätze für die Bildinterpretation

Im vorigen Abschnitt haben wir Verfahren vorgestellt, die für die Analyse und Interpretation von Gebäudebildern entwickelt wurden. Im Folgenden werden wir noch einige überwiegend hierarchische Ansätze zusammenstellen, die für allgemeine Aufgaben konzipiert wurden und häufig auf Tierbildern und Landschaftsaufnahmen getestet wurden bzw. auf Bildern, die technische Objekte wie Autos und Flugzeuge zeigen. Die meisten dieser Verfahren werden zur Kategorisierung von Bildern oder zu deren Segmentierung verwendet.

Die hierarchische Struktur eines Objekts und seinen Bestandteilen wird von Epshtein & Ullman (2005) und Epshtein & Ullman (2007) als Baum von rechteckigen Bildblöcken modelliert. Die Bildblöcke werden in verschiedenen Größen erzeugt und auf einem Trainingsdatensatz von Bildern bzgl. der Verwendbarkeit zur Objekterkennung analysiert. Die Hierarchie der Bildblöcke von Epshtein & Ullman (2005) wird auf der Basis von Überlappung erzeugt und enthält viele Bildblöcke, die zwar einen Ausschnitt des Objekts zeigen, aber in vielen Fällen sind die Bildblöcke nicht interpretierbar. Epshtein & Ullman (2007) minimieren die Anzahl der Bildblöcke in ihrer hierarchischen Struktur und konstruieren diese aus unkorrelierten Bildblöcken. So wählen sie viele Bildblöcke aus, die charakteristische Bestandteile eines Objekts zeigen, z. B. Ohr, Nase und Mund in Bezug auf Gesichter.

Bislang wurden diese Ansätze vor allem zur Erkennung von Tieren und Gesichtern eingesetzt, nicht aber zur Interpretation von Gebäudebildern, wo viele Objekte homogen oder gleichförmig texturiert erscheinen. Das erschwert die Auswahl der Bildblöcke. Zudem erschweren die vielen wiederholten Strukturen wie Fenster das Lernen der räumlichen Zusammenhänge zwischen den Bildblöcken. Lifschitz (2005) hat hierarchische Objekterkennung auf sehr kleinen und sehr stark geglätteten Fassadenbildern untersucht, aber eine Erkennung von Fassadenteilen war auf diesen Bildern auch für den Menschen nur begrenzt möglich.

Eine alternative Vorgehensweise bietet der Ansatz von Ommer (2007) und Ommer & Buhmann (2010), die die Bildblöcke an detektierte Bildpunkte koppelt und sukzessive zu größeren Aggregaten zusammensetzt. Dabei wird einerseits eine Hierarchie von Objektteilen konstruiert, anderseits werden für die Hierarchiebildung die Nachbarschaften zwischen nah bei einander liegenden Bildblöcken ausgewertet. Das Verfahren kann wegen der Initialisierung mit einem Punktoperator nur zur Erkennung texturierter Objekte oder von Objekten mit markanten Formen eingesetzt werden. So demonstrieren Ommer & Buhmann (2010) die Leistungsfähigkeit des Verfahrens an Flugzeugen, Schiffen und Tieren. Als weiteres Defizit des Verfahrens stellen die kleinen und aggregierten Teile nur selten semantische Bestandteile des Objekts dar, sondern kennzeichnen lediglich charakteristische Formen wie vorderer bzw. hinterer Bereich eines Objekts.

Pantofaru *et al.* (2008) benutzen für ihre Bildinterpretation verschiedene Bildsegmentierungen mit unterschiedlich gewählten Parametern. Aus den verschiedenen Bildpartitionen wird eine kombinierte berechnet, indem alle segmentierten Regionen miteinander geschnitten werden. So entstehen robuste Regionen und kleine Restregionen, die die Variabilität der Segmentierungsergebnisse darstellen. Die urspünglichen Regionen werden noch zur Extraktion geeigneter Merkmale und zum Lernen von Likelihood-Funktionen verwendet, die anschließend

gemäß der Schnittbildungen kombiniert werden. Das Verfahren kann unter Berücksichtigung von Nachbarschaftsbeziehungen zwischen den Segmenten analog zu den Arbeiten von Shotton *et al.* (2006) oder Verbeek & Triggs (2007) verbessert werden. Pantofaru *et al.* (2008) geben zu, dass es schwierig ist, aussagekräftige Merkmale für sehr kleine Regionen zu bestimmen. Deshalb arbeiten sie mit großzügig extrahierten Regionen. So bezweifeln wir einen erfolgreichen Einsatz dieses Verfahrens bei der Interpretation detailreicher Fassadenbilder und von Luftbildern, da hier viele kleine Regionen segmentiert werden.

Auch Hoiem *et al.* (2008) verwenden verschiedene Segmentierungsverfahren zur Bildinterpretation, schätzen zudem den Horizont der Szene und bestimmen in einem iterativen Verfahren Oberflächenbegrenzung, -orientierung und Entfernung der Objekte zur Kamera. In einfachen Landschaftsszenen erzielen Hoiem *et al.* (2008) gute Ergebnisse, aber die Segmentierungen der Stadtszenen ist nahezu unbrauchbar: Oft werden die Gebäude einer Straße als eine Region erkannt, die sich lediglich vom Himmel und der Straße abgrenzen. Eine detaillierte Bildinterpretation mit Angaben über die Anzahl und Anordnung von Fassadenteilen wie Fenstern ist so nicht möglich.

Gu *et al.* (2009) und Lim *et al.* (2009) nutzen die Bildsegmentierung von Arbeláez *et al.* (2009) für die Detektion einzelner Objekte in Bildern. Die hierarchische Segmentierung wurde zudem in (Arbeláez *et al.*, 2011) verbessert und ausführlicher dargestellt. Gu *et al.* (2009) nutzen diese Segmentierung zur Erstellung eines Baums relevanter Regionen, wobei die Relevanz durch ein Bag-of-Words-Konzept entschieden wird, d. h. zuvor wird gelernt, welche Bildausschnitte für die Erkennung von Objekten relevant sind. Der Fokus ist dabei wie auch in anderen Arbeiten die Erkennung des gesamten Objekts, nicht aber die Identifikation seiner Bestandteile. Lim *et al.* (2009) greifen die Hierarchie der Regionen auf, um so für die Regionen der untersten Hierarchiestufe lokale, regionale und globale Merkmale zu bestimmen. Eine Erkennung komplexer Objekte ist damit erst möglich, wenn gleichklassifizierte Regionen zusammengefasst werden. Allerdings bleibt auch hier die Analyse der Bestandteile aus.

Graphische Modelle finden bei der Bildinterpretation seit einigen Jahren eine große Akzeptanz, weil sie eine gute Möglichkeit darstellen, Kontext in die Szene zu integrieren. Besonders erfolgreich sind dabei Markoff Zufallsfelder für die Integration von Nachbarschaftsbeziehungen, Bayes-Netze für die Integration von hierarchischen Beziehungen und hierarchische Zufallsfelder, die die Nachbarschaftsinformation auf verschiedenen Maßstabsebenen auswerten und so hierarchische und nachbarschaftliche Information vereinigen. Ansätze dazu stammen z. B. von Jin & Geman (2006), Schnitzspan *et al.* (2008) und Plath *et al.* (2009).

Jin & Geman (2006) haben ihr graphisches Modell für die Erkennung von Nummernschildern designt, dessen Struktur manuell bestimmt wurde. Die hierarchische Modellierung wurde hierbei von den Bayes-Netzen inspiriert und besteht aus diversen semantischen Ebenen, in denen sukzessive die Bestandteile des Nummernschilds erkannt und zusammengesetzt werden. Eine Adaption des Verfahrens für die Interpretation von Gebäudebildern ist unrealistisch. Die Struktur des graphischen Modells ist viel zu komplex und müsste zudem automatisch bestimmt werden können.

Schnitzspan *et al.* (2008) zerlegen das Bild in ein regelmäßiges Gitter von aneinandergrenzenden Bildblöcken, um so Regionen zu definieren, und einen einfachen Nachbarschaftsgraphen. Eine hierarchische Struktur erzeugen Schnitzspan *et al.* (2008) zudem durch das Zusammenlegen von vier Blöcken, wie es auch in regelmäßigen Bildpyramiden realisiert wird. In jeder Ebene wird nun ein Bedingtes Zufallsfeld modelliert, durch die verschiedenen Ebenen können so Merkmale der Blöcke und zugehörige Nachbarschaftsbeziehungen aus verschiedenen

Maßstabsebenen gewonnen werden. Einen analogen Ansatz mit allgemeinen Bildsegmentierungen aus mehreren Maßstabsebenen haben Plath *et al.* (2009) realisiert. Das Lernen der binären Potentialfunktionen der Bedingten Zufallsfelder ist sehr aufwendig. Mit einem ähnlichen Modell sind Yang *et al.* (2010) beim Lernen der Wahrscheinlichkeiten aus Komplexitätsgründen gescheitert.

1.3. Arbeitshypothese

Die hierarchische Zerlegung eines Objekts in seine Bestandteile möchten wir in dieser Arbeit nutzen, um die Bildinterpretation zu verbessern. Durch geeignete Verfahren können wir komplexe Objekte wie Häuser und ihre Bestandteile in einem einzelnen Bild detektieren und sie in einer hierarchischen Struktur von detektieren Bildbereichen anordnen. Diese Baum-Struktur ist ferner dazu geeignet, die Interpretation der detektierten Bereiche zu verbessern, da man von erkannten kleinen Teilen auf zusammengesetzte Teile schließen kann und von den zusammengesetzten Aggregaten auf ihre Bestandteile. Dieser Schließungsprozess kann wegen der hierarchischen Struktur der detektierten Bildregionen durch ein Bedingtes Bayes-Netz realisiert werden. Konkret formulieren wir die folgenden vier Hypothesen für diese Arbeit:

1. Es ist möglich, eine hierarchische Bildsegmentierung zu erzielen, in der komplexe Objekte und ihre Bestandteile detektiert werden können.

2. Die Relationen der Bestandteilshierarchie des Objekts bilden sich in prädizierbarer Weise in die Hierarchie der Segmentierung ab.

3. Für die Bildbereiche der hierarchischen Segmentierung können aussagekräftige Merkmalsvektoren bestimmt werden, die eine hinreichend gute Vorklassifikation dieser Elemente ermöglichen.

4. Die Interpretation durch ein Bedingtes Bayes-Netz, dessen Struktur durch die hierarchische Segmentierung definiert wird, verbessert das Ergebnis der Vorklassifikation.

Die detektierten Bildbereiche werden bei der hierarchischen Bildsegmentierung in einer Baum-Struktur angeordnet, so dass durch diese Relation Zugehörigkeiten von Teilen zu Aggretgaten repräsentiert werden. Der Detektionserfolg hängt auch von einer benutzerdefinierten Ontologie ab, in der definiert wird, welche Objektteile bei der Bildinterpretation beachtet werden sollen. Von dieser Ontologie für den Objektraum wird eine Ontologie für die Bildinterpretation abgeleitet, die zusätzlich die gegebenen Aufnahmeverhältnisse berücksichtigt. Ein wichtiger Punkt ist dabei die Bildauflösung: Wenn ein Objektteil nur auf ein paar Pixel abgebildet wird, dann reicht die Bildauflösung eventuell nicht aus, dieses Objekt zu erkennen. Das ist z. B. bei Luftbildern der Fall, in denen Schornsteine und andere kleine Dachaufbauten wie Antennen fast nicht sichtbar sind. Ein weiterer Punkt ist der Blickwinkel auf das Objekt: In der Mitte von Luftbildern sind die Seitenwände von Gebäuden nicht sichtbar, in terrestrischen Innenstadtaufnahmen sieht man oft das Dach nicht.

Verdeckungen und schwache Kontraste im Bild stellen bei der datengetriebenen, hierarchischen Bildsegmentierung ein großes Problem dar. Nach der Projektion in die Bildebene können die Abbildungen einzelner Objekte bzw. deren Teile benachbart sein, die im Objektraum nicht benachbart sind. Ein graues Hausdach und eine gepflasterte Straße können sehr ähnlich aussehen und durch die Nachbarschaft im Bild als homogener Bildbereich und damit als ein Bildsegment erkannt werden. Das führt dann zu Relationen in der Hierarchie der Segmentierung, die nicht den Relationen der Bestandteilshierarchie entsprechen.

Die ersten beiden Hypothesen stellen die Grundlage für die Bildinterpretation dar. Hier werden die zu interpretierenden Bildbereiche bestimmt und die hierarchische Struktur, die zu einem besseren Klassifikationsergebnis führen soll. Wenn wir davon ausgehen, dass ein Fenster im Dach dadurch besser als solches erkannt wird, weil wir in einer höheren Segmentierungsebene das Dach erkannt haben, dann muss die Struktur der detektierten Bildbereiche diese Bestandteilshierarchien widerspiegeln. Wir modellieren diese Struktur als Baum, weil es einerseits der Modellierung im Objektraum entspricht, andererseits kann diese Baum-Struktur in der weiteren Arbeit für ein Bedingtes Bayes-Netz verwendet werden.

Wir verwenden eine überwachte Klassifikation für die Bestimmung der Klassenzugehörigkeiten der detektierten Bildbereiche. Dazu verwenden wir manuell erstellte Annotationen der Bilder, mit denen jedem detektierten Bildbereich das bestmögliche Klassenlabel zugewiesen wird. Ferner definieren wir Merkmale, die einerseits die Form der Region charaktirisieren, aber auch ihre Erscheinung, teilweise auch im Kontrast zur Umgebung. Eine Klassifikation der Bildbereiche anhand dieser Merkmale ist dann hinreichend gut, wenn jedem Bildbereich einer Klasse zugewiesen wird und diese Klassifikation signifikant besser ist als der Zufall bzw. besser ist, als wenn alle Bildbereiche der am häufigsten auftretenden Klasse zugewiesen werden.

Die Hierarchie von detektierten Bildbereichen ermöglicht die Integration von Kontext. Das Ergebnis der Vorklassifikation wird mit den Beobachtungen in der Hierarchie zu einer neuen Interpretation der Bildbereiche vereint. Dazu leiten wir aus der Baum-Struktur der detektierten Bildbereiche ein Bedingtes Bayes-Netz ab. Ein Knoten dieses Bayes-Netzes stellt eine Zufallsvariable dar, die die Wahrscheinlichkeiten für die Klassenzugehörigkeiten des entsprechenden Bildbereichs angibt.

1.4. Gliederung der Arbeit

In Kapitel 2 stellen wir unser Konzept für die hierarchische Interpretation von Bildern vor und führen die in dieser Arbeit verwendete Notation ein. Die Graphen-Struktur des Bayes-Netzes wird aus der hierarchsichen Segmentierung des Bildes gewonnen, die Zufallsvariablen des Netzes geben die Wahrscheinlichkeiten an, mit der der detektierte Bildbereich zu einer bestimmten Klasse gehört, und die Übergangswahrscheinlichkeiten bestimmen wir unter Verwendung der manuell annotierten Referenzbilder. Angaben zur Evaluation unseres Konzepts und eine Beschreibung der verwendeten Datensätze schließen das zweite Kapitel ab.

Die Umsetzung unseres Konzepts wird in den drei anschließenden Kapiteln erläutert. In Kapitel 3 thematisieren wir die hierarchische Segmentierung mit dem Wasserscheidenalgorithmus, d. h. den Aufbau einer irregulären Bildpyramide sowie die Fokussierung auf stabile Elemente in dieser Pyramide. Die Klassifikation der segmentierten Bildbereiche besprechen wir in Kapitel 4, wobei vor allem auf die verwendeten Merkmale und das Design des Mehrklassen-Klassifikators, die Alternierenden Entscheidungsbäume, eingehen. Beide Komponenten, die hierarchische Segmentierung und die Klassifikation der Region auf Basis ihrer Merkmale, werden im Bedingten Bayes-Netz zusammengeführt, worauf wir in Kapitel 5 eingehen. Alle drei Kapitel werden jeweils mit der Präsentation von Ergebnissen abgeschlossen.

Im sechsten und letzten Kapitel fassen wir die entwickelte Methode dieser Arbeit zusammen, diskutieren die erzielten Ergebnisse und geben Anregungen für Verbesserungen bzw. weiterführende Arbeiten.

2. Konzept für die hierarchische Interpretation von Bildern

2.1. Bayes-Netz für die hierarchische Interpretation

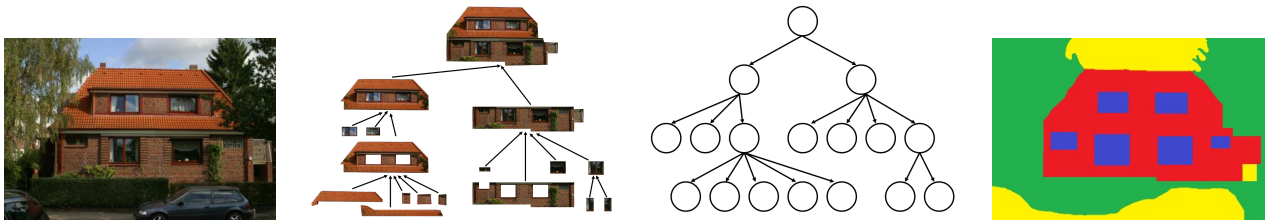
In diesem Abschnitt stellen wir unser Konzept für die hierarchische Bildinterpretation durch ein Bayes-Netz vor. In Abb. 2.1 zeigen wir die wichtigen Bestandteile des Konzepts: V. l. n. r. zeigen wir zuerst das gegebene Bild eines Gebäudes, dann stellen wir die hierarchische Struktur der segmentierten Regionen im Bereich des Gebäudes dar, aus der wir das Bedingte Bayes-Netz konstruieren, und als Ausgabe zeigen wir Klassifikationsbilder an, in denen die Bildbereiche markiert werden, die als Instanzen einer Klasse erkannt wurden.

Wir werden in diesem Abschnitt die Notation für die wichtigsten Komponenten einführen. Dazu beginnen wir bei den Klassen, gehen dann auf die Referenzobjekte, d. h. die manuell gekennzeichneten Annotationen, ein, und stellen dann die Notation für die automatisch segmentierten Regionen vor. Zum Schluss definieren wir den Begriff *Bedingtes Bayes-Netz*. Dabei gehen wir auch auf die zu lernenden Wahrscheinlichkeiten ein, die durch das Bayes-Netz modelliert werden.

2.1.1. Ontologie der Objektklassen

Unter einer Bildinterpretation verstehen wir eine symbolische Bildbeschreibung, in der erkannte Objekte gekennzeichnet werden. Die Erkennung von Objekten setzt die Defintion von Objektklassen voraus, d. h. eine Bezeichnung sowie eine Beschreibung für jede Klasse. Wir haben die Definition dieser Klassen in einer anwendungsspezifischen Ontologie umgesetzt.

Die Menge der in der Ontologie definierten Klassen bezeichnen wir mit Ω'. Um die Kardinalität dieser Menge auf eine überschaubare Größe K' begrenzen zu können, haben wir zusätzlich eine Klasse Hintergrund, `background`, eingefügt. Zu dieser Objektklasse gehören alle Objekte, die durch keine der anderen $K' - 1$ Klassen erfasst werden. Nach der Projektion der Objekte in die Bildebene verwenden wir diese Klassen auch zur Annotation von Bildern.



(a) (b) (c) (d)

Abbildung 2.1.: Komponenten des Konzepts für die hierarchische Bildinterpretation: (a) Eingabe eines Farbbilds, (b) Aufbau einer hierarchischen Struktur von segmentierten Regionen (c) Konstruktion des Bayes-Netzes sowie (d) Ausgabe von Klassifikationsbildern für Gebäude bzw. Fenster.

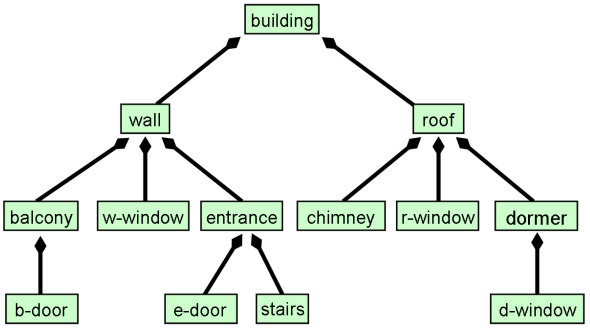


Abbildung 2.2.: Bestandteilshierarchie aus Ontologie.

Neben der Menge der Klassen werden in Ontologien auch die mereologischen, topologischen sowie taxonomischen Beziehungen zwischen Instanzen dieser Klassen beschrieben, siehe Dörschlag *et al.* (2008). Mereologische Beziehungen geben die Bestandteilshierarchien von Objekten wieder, topologische Beziehungen Nachbarschaftsverhältnisse und taxonomische Beziehungen Spezialisierungen. In Bezug auf die Modellierung von Gebäuden sind mereologische Beziehungen z. B. die Zerlegung eines Gebäudes in Wände und Dach, in den Wänden befinden sich wiederum Fenster und Türen. Dass ein Dach immer oben an die Wände eines Hauses grenzt oder ein Fenster eine Öffnung, d. h. ein Loch in der Wand darstellt, sind typische topologische Beziehungen. Eine taxonomische Beziehung liegt vor, wenn man z. B. bei Fenstern zwischen den Fenstern einer Fassadenwand und den Fenstern eines Daches unterscheidet.

Mögliche mereologische Beziehungen für die Modellierung von Gebäuden in Bildern werden in Abb. 2.2 gezeigt, wobei die Pfeile *ist Teil von* ausdrücken. Die zusätzliche Spezialisierung von Fenstern und Türen macht den Graphen übersichtlich und gibt ihm eine baumartige Struktur. Bei der Bildanalyse bestimmen wir die mereologischen Beziehungen zwischen detektierten Bildausschnitten durch eine hierarchische Segmentierung, wobei die detektierten Bildausschnitte zu sukzessive größer werdenden Bildbereichen zusammengefasst werden. Diese Gruppierungen können fehlerhaft sein. Einerseits können falsche Zuordnungen vorkommen, d. h. in der Hierarchie werden zwei Bildausschnitte zusammengeführt, die nicht zusammengehören. Andererseits können aber auch Teile fehlen, weil sie durch andere Objekte verdeckt werden und somit im Bild nicht sichtbar sind.

Bei den topologischen Relationen können im Bildraum Nachbarschaften zwischen abgebildeten Objektteilen auftreten, die es im Objektraum nicht gibt. So kommt es in Luftbildern z. B. zu Nachbarschaftsbeziehungen zwischen einem projizierten Dach und dem Boden, auf dem das Haus errichtet wurde. Abb. 2.3 zeigt vier verschiedene Ansichten eines Gebäudes mit Balkon. Der Balkon befindet sich an einer der vier Außenwände des Gebäudes, die wie die Dachflächen des Pyramidendachs und die Seiten des Balkons durch Drei- bzw. Vierecke dargestellt werden. In allen vier Ansichten kommen verschiedene Nachbarschaftsbeziehungen zwischen den Polygonen zum Vorschein, die so nicht im Objektraum modelliert werden.

Diese Arbeit will die Unterstützung der Bildinterpretation durch Auswertung der detektierten Bestandteilshierarchien untersuchen. Somit fokussieren wir uns auf die Integration mereologischer Beziehungen in die Bildinterpretation. Wenn wir zudem die topologischen Beziehungen aus Komplexitätsgründen vernachlässigen, dann liefert die Hierarchie der detektierten Bildausschnitte eine Struktur, die für eine Bildinterpretation mit einem Bayes-Netz geeignet ist. Nachbarschaften zwischen den segmentierten Bildbereichen werden im Bayes-Netz nicht explizit modelliert. Einen Vorschlag zur Erweiterung des Konzepts, d. h. zur Einbindung der

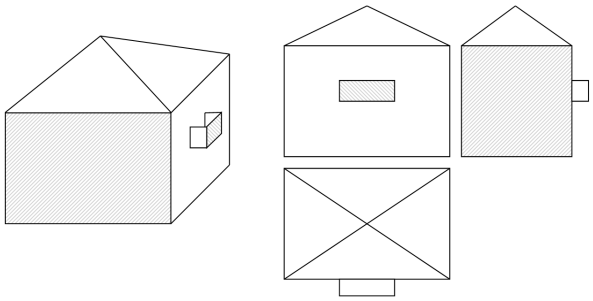


Abbildung 2.3.: Vier verschiedene Ansichten eines Gebäudes mit Balkon.

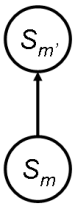


Abbildung 2.4.: Hier zeigen wir schematisch zwei hierarchisch angeordnete Regionen S_m und $S_{m'}$. Den Pfeil haben wir analog zur Bestandteilshierarchie ausgerichtet, die wir im Zusammenhang mit der Ontologie definierten. Die Elternbeziehung zwischen beiden Regionen wird in unser Notation durch $m' = \pi(m)$ ausgedrückt. Entsprechnend zeigt $m = \chi_i(m')$ an, dass die Region S_m das i-te Kind der Region $S_{m'}$ ist.

taxonomischen Relationen mit Markoff-Zufallsfeldern geben wir im Ausblick im letzen Kapitel dieser Arbeit.

2.1.2. Annotationen und Regionen

Das Trainieren von Klassifikatoren und die Evaluierung des Konzepts dieser Arbeit erfolgen durch Vergleich mit Referenzdaten. Diese haben wir manuell erstellt und bezeichnen sie als Annotationen. Die Menge der Annotationen geben wir mit $\mathcal{T}$ an, zu der L Elemente gehören, die wir T_l nennen. Wir verstehen darunter

1. die geometrische Ausprägung eines Bildbereichs, die durch einen Polygonzug charakterisiert wird,
2. die Zuweisung einer Klasse ω_k zu diesem Bildbereich sowie
3. eine Liste von Indices auf andere Annotationen zur Angabe der Bestandteilshierarchie.

Als Regionen bezeichnen wir automatisch detektierte Bildbereiche, die wir mit $S_m \in \mathcal{S}$ bezeichnen. Die M Elemente sind hierarchisch angeordnet: Die eindeutige Elternregion wird durch die Funktion π, die möglichen Kinderregionen durch die Funktion χ bestimmt, siehe Abb. 2.4. Beide Funktionen arbeiten auf den Indices der Regionen und geben auch einen Index zurück, bei mehreren Kindern werden die Indices durchnummeriert. Beide Funktionen sind außerdem auf die Annotationen übertragbar, um auch dort die Bestandteilshierarchie mathematisch ausdrücken zu können.

Für unser Konzept hat die Regionenhierarchie eine große Bedeutung, da sie die Struktur für das Bayes-Netz bildet. In Abb. 2.5 zeigen wir eine solche Hierarchie. Auf den ersten Blick sieht die Auswahl von Regionen als auch deren hierarchische Anordnung sehr zufriedenstellend aus. Dennoch zeigt die Abbildung auch viele Problemfälle, die durch die Segmentierung

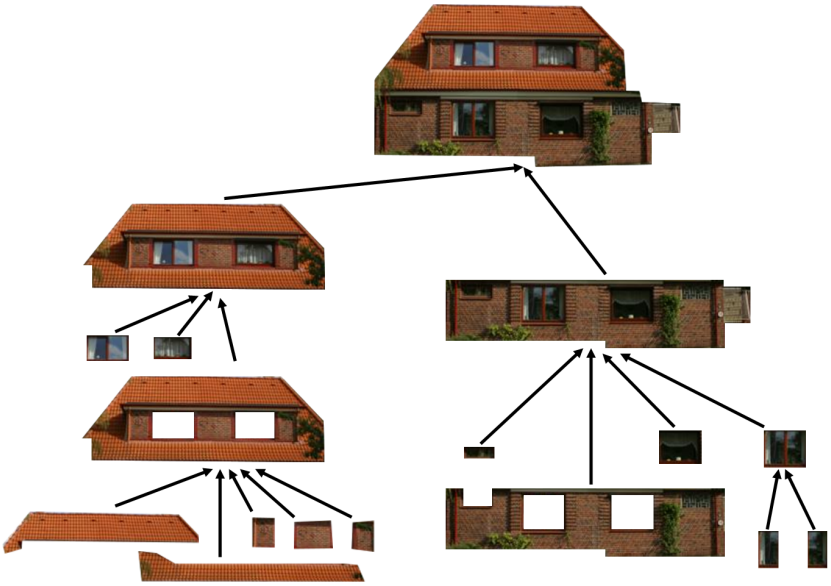


Abbildung 2.5.: Hierarchische Segmentierung einer Gebäudeszene.

verursacht werden. Erstens zeigt die das gesamte Haus repräsentierende Region auch viele Pixel, die Vegetation darstellen. Zweitens gibt es Regionen, die nur einen Teil des Objekts darstellen, z. B. die Dachfläche oder die Gaubenaußenwand in der untersten Hierarchiebene. Und drittens gibt es keine Region, die die vollständige Gaube auf dem Dach repräsentiert. Die Wand-Regionen der Gaube werden in der folgenden Hierarchieebene direkt mit den andern Dachregionen zusammengefasst statt mit den Fenstern zur Gaube. Der erste Fall motiviert uns zur Erweiterung der Menge der Klassen Ω' (siehe Kap. 2.1.3), die anderen beiden Fälle zeigen die Problematik mit Über- bzw. mit Untersegmentierung.

2.1.3. Klassenlabel der Regionen

Da wir eine überwachte Klassifikation der Regionen durchführen, um eine möglichst gute Performanz des Bayes-Netzes zu erhalten, benötigen wir eine Zuordnung von Regionen und Klassen. Wir bezeichnen das Klassenlabel der Region S_m mit $\widetilde{x}_m$.

Die Menge Ω' enthält alle Klassen, mit denen 3D Objekte und ihre Abbildungen im 2D gemäß der gegebenen Ontologie benannt werden können. Da automatisch detektierte Regionen nicht notwendigerweise die Abgrenzungen der abgebildeten Objekte haben müssen, d. h. sie zeigen nur einen Ausschnitt eines Objekts oder mehrere verschiedene Objekte zusammen, wird die Menge Ω' erweitert, und es entsteht eine Menge Ω mit $\mathcal{K}$ Klassen ω_k für die Bezeichnung von Regionen.

Die Erweiterung der Klassen wird durch drei Fälle motiviert, die wir in Abb. 2.6 darstellen. Links ist der Idealfall zusehen, wo die segmentierte Region, dargestellt durch eine schwarze Ellipse mit durchgezogenem Rand, sehr gut zur manuellen Annotation, dargestellt durch ein rotes Rechteck mit gestricheltem Rand, passt. Die blaue gepunktete Ellipse im rechten Teil zeigt dagegen eine Region, die nur einen kleinen Teil einer Annotation zeigt. Wenn diese Region z. B. Himmel oder Vegetation zeigt, so kann die Region das entsprechende Klassenlabel übernehmen. In diesen Fällen ist die Annotation nicht unbedingt an ein Objekt mit einer klar sichtbaren Begrenzung gekoppelt. Bei Fassaden und Dachflächen dagegen gibt es kleine

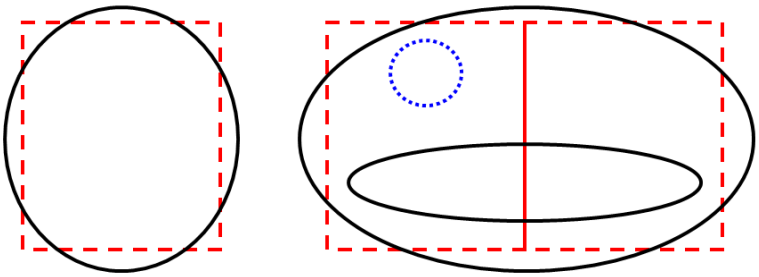


Abbildung 2.6.: Manuell erstellte Polygonzüge der Annotation werden als rote, gestrichelte Rechtecke dargestellt, automatisch segmentierte Regionen als Ellipsen, die einen blauen, gepunkteten bzw. einen schwarzen, durchgezogene Rand haben. Links zeigt die Abbildung einen Fall, in dem Region und Annotation gut zusammenpassen. Dort weisen wir der Region die Klasse der Annotation zu. Im rechten Teil sind die Fälle dargestellt, die eine gute Klassenzuweisung schwierig gestalten, weshalb wir Teilklassen und die neue Klasse `mixture` eingeführt haben.

Teilobjekte in einem deutlich größeren Objekt. Da die Wände und Dachflächen ohne diese Teile nicht extra annotiert wurden, in ihnen aber häufig kleine Regionen gefunden werden, haben wir zwei neue Teilklassen `facade-tiles` und `roof-tiles` für diese Bereiche eingeführt.

Die beiden schwarzen Ellipsen auf der rechten Seite von Abb. 2.6 überlappen zwei Annotation zu gleichen Teilen. Da beide Regionen zwei Objekte zumindestens teilweise zeigen, stellen sie Mischungen dieser Objekte oder Objektteile dar. Das motiviert uns zur Einführung des Klassenlabels `mixture`. Eine Unterteilung der beiden Fälle in Mischklasse von ganzen Objekten oder von Teilen sowie eine weitere Unterteilung nach Klassen, z. B. Fassade-Vegetation oder Vegetation-Boden, haben wir aus Komplexitätsgründen unterlassen. Allerdings führen wir die drei Spezialklassen `facade-mixture`, `roof-mixture` und `building-mixture` ein, um Regionen ein gutes Klassenlabel geben zu können, wenn sie mehrere Teile einer Fassade, eines Daches oder des Gebäudes, aber nicht das gesamte Objekt darstellen.

Im Folgenden wird der Algorithmus zu Bestimmung des besten Klassenlabels $\widetilde{x}_m$ der Region S_m beschrieben. Die Zuordnung erfolgt durch Vergleich der Region mit allen L manuell erstellten Annotationen T_l. Gebe $|S_m|$ den Flächeninhalt der Region an, dann führen die folgenden drei Entscheidungen zu einem geeigneten Klassenlabel $\widetilde{x}_m$. Sie können durch einen Entscheidungsbaum repräsentiert werden, siehe Abb. 2.7. Die Wahl der Schwellwerte basiert auf stichprobenartigen Überprüfungen.

1. Der erste Test untersucht, ob sich die Region S_m und annotierte Bildbereiche T_l überlappen. Wenn die Region kaum annotierte Bildbereiche schneidet, d. h.

$$\frac{|S_m \cap \left(\bigcup_{l=1}^{L} T_l\right)|}{|S_m|} < \theta_1,$$

   dann erhält sie das Klassenlabel `background`.

2. Anderenfalls wird die Annotation T_{l*} gesucht, die am besten zur Region passt, d. h.

$$l^* = \arg\max_l \frac{|S_m \cap T_l|}{|S_m \cup T_l|}.$$

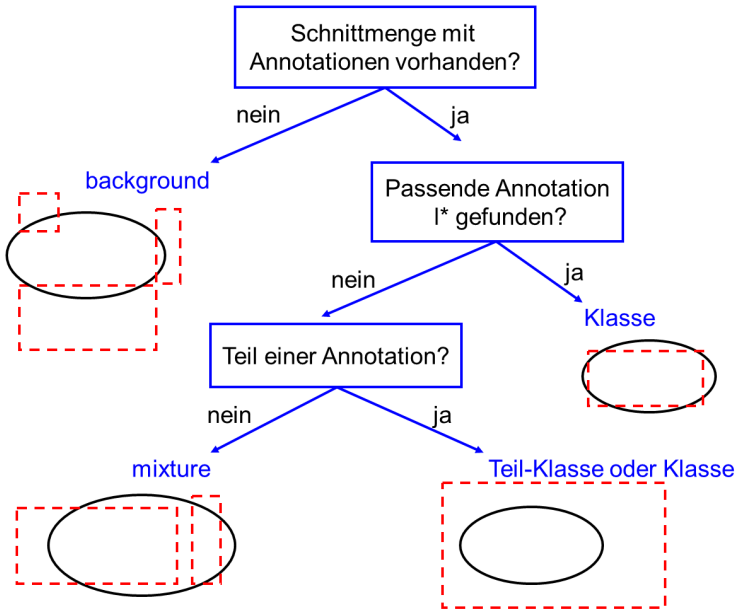


Abbildung 2.7.: Drei Entscheidungen führen zur Wahl des am besten passenden Klassenlabels. Der Entscheidungsbaum visualisiert die Reihenfolge der Analyse bzgl. der vorliegenden Konfiguration zwischen der schwarzen ellipsenförmigen Region und den roten rechteckförmigen Annotationen.

Wenn

$$\frac{|S_m \cap T_{l*}|}{|S_m \cup T_{l*}|} > \theta_2,$$

dann haben die Region und die beste Annotation eine große gemeinsame Schnittmenge und ihre Vereinigung ist nur unwesentlich größer. In diesem Fall erbt die Region S_m das Klassenlabel der Annotation T_{l*}.

3. Anderenfalls sind noch zwei Fälle zu unterscheiden. Entweder schneidet die Region S_m mehrere Annotationen, oder sie Teil einer deutlich größeren. Für diese Entscheidungen untersuchen wir, welche Klassen die Annotationen haben, die einen großen Schnitt mit der Region haben. Je nachdem, welche Klassen es sind und ob diese Klassen in einer Teil-von-Beziehung stehen, wird dann das am besten geeignete Klassenlabel bestimmt. Im ersten Fall stellt die Region eine Mischklasse dar, wobei wir noch unterscheiden, ob es sich bei den Annotationen um gleichberechtigte Objekte oder um ein Objekt und seine Teile handelt. So erhält die Region dann entweder das allgemeine Klassenlabel `mixture` oder ein spezielleres, z. B. das Klassenlabel `building-mixture`. Im anderen Fall, wenn die Region nur einen Teil einer Annotation T_l zeigt, erhält sie entweder das Klassenlabel dieser Annotation (wenn sich das Objekt nicht sinnvoll teilen lässt, z. B. Himmel), oder die Region erhält ein spezielles Teilklassen-Label, z. B. `roof-tiles` für die kleineren Dachschindeln.

Die relativen Häufigkeiten der Klassenlabel werden sich stark unterscheiden. Das liegt einerseits an den verschiedenen relativen Häufigkeiten der Objekte als auch deren Größe im Bild. So ist zuerwarten, dass Gebäuderegionen und Fensterregionen signifikant häufiger vorkommen werden als z. B. Tür- oder Schornsteinregionen. Dieses Ungleichgewicht zwischen den Klassen

wird sich auch auf die Klassifikation auswirken. Dem können wir entweder durch eine Bewertung der Klassifikationen entgegentreten, wie sie auch bei der Risikominimierung angewendet wird, d. h. im Training wird die Fehlklassifikation bei einer Türregion stärker bestraft als die Fehlklassifikation einer Gebäuderegion, vgl. (Duda *et al.*, 2001). Oder wir legen einzelne Klassen nachträglich zusammen. Dieses kann z. B. inhaltlich entschieden werden, weil sich die Klassen stark ähneln.

Wir haben uns für die zweite Möglichkeit entschieden, da wir anderenfalls Bewertungskriterien für die Fehlklassifikationen erstellen müssen. Beispielsweise können wir die beiden Klassen `road` und `pavement` im Benchmark-Datensatz `terra-1` semantisch zusammenlegen oder die beiden Klassen `window` und `window-pane` im originalen eTRIMS-Datensatz `terra-2`. Andere Klassen wiederum können wir auf diese Weise nicht mit anderen zusammengelegen, z. B. passt die Klasse `sky` weder zur Klasse `vegetation` noch anderen Klassen wie `road` oder `window` etc.

Somit führen wir eine neue Klasse `others` ein, in der wir alle selten vorkommenden Regionen (gemessen an der Anzahl der Regionen mit entsprechendem Klassenlabel) vereinen. Da diese sehr unterschiedliche Klassen wir z. B. Himmel, Boden und Türen vereinen wird, halten wir eine geschachtelte Klassifikation für realistisch. Im ersten Durchgang klassifizieren wir K Klassen, anschließend folgt ein zweiter Klassifikationschritt zur Trennung der Teilklassen von `others`. Da wir letzteres aus Zeitgründen nicht mehr realisiert haben, geben wir die Ergebnisse der Bildinterpretation, d. h. der Klassifikation der Regionen durch das Bedingte Bayes-Netz, mit K statt mit K Klassen an.

2.1.4. Bedingtes Bayes-Netz

Die Klassifikation der Regionen S_m wollen wir durch ein Bedingtes Bayes-Netz verbessern. Daher werden wir in diesem Abschnitt unser verwendetes Graphisches Modell, das Bedingte Bayes-Netz, motivieren. Zunächst werden wir dazu Bayes-Netze allgemein einführen, dann den Bezug zu Markoff-Zufallsfeldern herstellen, um abschließend Bedingte Bayes-Netze als hierarchische Bedingte Markoff-Zufallsfelder einführen zu können.

2.1.4.1. Bayes-Netze

Als ein Vertreter der Graphischen Modelle werden Bayes-Netze als gerichtete, azyklische Graphen definiert. Jeder Knoten eines Bayes-Netzes stellt eine Zufallsvariable x_m, die einen von K_m Zuständen annimmt. Durch die gerichteten Kanten von $x_{\pi(m)}$ nach x_m wird die Abhängigkeit der Zufallsvariable x_m von $x_{\pi(m)}$ modelliert. $x_{\pi(m)}$ ist dann Elternknoten von x_m. Zu diesen gerichteten Kanten werden Übergangswahrscheinlichkeiten bestimmt, die diese Abhängigkeiten einer Zufallsvariable von anderen in Form bedingter Wahrscheinlichkeiten wiederspiegeln. In Abb. 2.8 zeigen wir ein Beispiel für ein Bayes-Netz mit sieben Zufallsvariablen. Die drei Zufallsvariablen x_1, x_4 und x_5 haben keine eingehenden Kanten, stellen somit unabhängige Zufallsvariablen dar. Dem gegenüber stehen die abhängigen Variablen, beispielsweise hängt der Zustand von x_6 nur von den Zufallsvariablen x_2, x_3 und x_5 ab und nicht von den anderen.

Durch die explizite Modellierung von Abhängigkeiten der Zufallsvariablen untereinander ermöglichen Graphische Modelle die Reduzierung der Komplexität für die Berechnung der gemeinsamen Wahrscheinlichkeitsverteilungen. Am Beispiel des Bayes-Netzes aus Abb. 2.8 bedeutet das, dass die gemeinsame Wahrscheinlichkeitsverteilung durch folgende Terme mo-

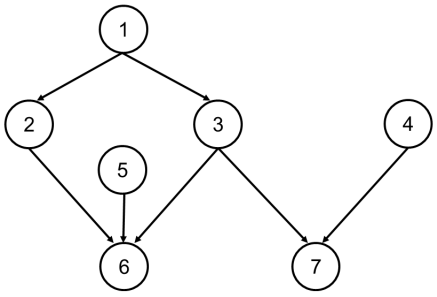


Abbildung 2.8.: Bayes-Netz mit sieben Zufallsvariablen.

delliert wird:

$$P(x_1, \ldots, x_7) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1)P(x_4)P(x_5)P(x_6 \mid x_2, x_3, x_5)P(x_7 \mid x_3, x_4). \quad (2.1)$$

Werden die Zufallsvariablen ohne Information über gegenseitige Abhängigkeiten modelliert, beschreibt man die gemeinsame Wahrscheinlichkeitsverteilung durch eine Tabelle mit $\mathsf{K}^{\mathsf{M}} = \mathsf{K}^{7}$ vielen Kombinationen von Zustandsbelegungen der Variablen. Die rechte Seite von Gl. 2.1 gibt dagegen Aufschluss über gegenseitige Wechselwirkungen der Zufallsvariablen, weshalb die gemeinsame Wahrscheinlichkeitsverteilung mit drei Wahrscheinlichkeiten einzelne Variablen betreffend sowie vier bedingten Wahrscheinlichkeiten modelliert wird, die jeweils K bis K^3 viele Kombinationen von Zustandsbelegungen von 1 bis 3 Variablen als Bedingungen enthalten. Somit ist ein entscheidender Nutzen von Bayes-Netzen, dass die gemeinamen Wahrscheinlichkeitsverteilungen nicht als exponentiell große Tabellen bestimmt werden, sondern in mehrere kleine in K quadratisch oder kubisch große von einander unabhängige Wahrscheinlichkeitsverteilungen zerlegt werden.

2.1.4.2. Moralisierung von Bayes-Netzen

Ein anderer wichtiger Vertreter der Graphischen Modelle sind die Markoff-Zufallsfelder mit ungerichteten Kanten, mit denen häufig symmetrische Abhängigkeiten bzw. Einflüsse zwischen den Zufallsvariablen wie Nachbarschaften formalisiert werden. Ein Bayes-Netz kann in ein Markoff-Zufallsfeld überführt werden, wenn man zunächst den Graphen moralisiert und anschließend die Richtung der Kanten entfernt (Bishop, 2006). Unter einer Moralisierung eines Graphen versteht man, dass alle Elternknoten einer Zufallsvariable durch hinzugefügte Kanten miteinander verbunden werden: Die Eltern werden 'verheiratet'. In Abb. 2.9 zeigen wir den moralisierten Graphen des Bayes-Netzes aus Abb. 2.8. Im Bayes-Netz hatte die Zufallsvariable x_6 die drei Elternknoten x_2, x_3 und x_5, die im moralisierten Graphen durch gestrichelte Kanten paarweise miteinander verbunden sind. Auch die beiden Eltern von x_7 wurden miteinander verheiratet.

Durch die Moralisierung des Graphen entstehen Cliquen, d. h. Teilmengen von Knoten, die alle miteinander durch Kanten verbunden sind. Jeder Clique wird eine Potentialfunktion zugeordnet wird, die große Werte annimmt, wenn die Konfiguration wahrscheinlich ist. So ergibt sich für die gemeinsame Wahrscheinlichkeitsverteilung im Markoff-Zufallsfeld bzgl. des Beispiels aus Abb. 2.8 ein Produkt der Cliquenpotentiale

$$P(x_1, \ldots, x_7) = \frac{1}{\zeta}\psi_{1,2}\psi_{1,3}\psi_{2,3,5,6}\psi_{3,4,7} \quad (2.2)$$

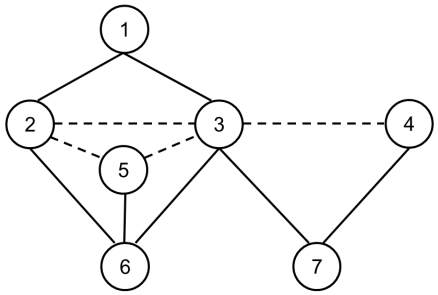


Abbildung 2.9.: Moralisierter Graph des Bayes-Netzes aus Abb. 2.8. Die gestrichelten Kanten wurden neu hinzugefügt, um die Elternknoten der Zufallsvariablen x_6 bzw. x_7 zu verheiraten.

mit z. B.

$$\begin{aligned} \psi_{1,2} &= P(x_1)P(x_2 \mid x_1) \\ \psi_{1,3} &= P(x_3 \mid x_1) \\ \psi_{2,3,5,6} &= P(x_6 \mid x_2, x_3, x_5) \\ \psi_{3,4,7} &= P(x_7 \mid x_3, x_4) \end{aligned} \tag{2.3}$$

Im Folgenden verwenden wir nur baumartige Bayesnetze, sodass die Cliquenbildung durch die Moralisierung nicht erforderlich ist, da jeder Knoten nur einen Elternknoten hat.

2.1.4.3. Bedingte Markoff-Felder und Bedingte Bayes-Netze

Bei Markoff-Zufallsfeldern basieren die Potentialfunktionen auf lokal extrahierten Merkmalen. Das kann zu Fehlinterpretationen der Szene führen, wenn sich die die Bestandteile verschiedener Objekte bei kontextfreier Betrachtung stark ähneln. Daher haben Lafferty *et al.* (2001) Markoff-Felder mit global extrahierten Merkmalen definiert, die Bedingten Zufallsfelder. Sie kennzeichnet, dass alle Zufallsvariablen und alle Potentialfunktionen unter der Bedingung von beobachteten Daten $\boldsymbol{F}$ formuliert werden. Somit wird nun das Maximum der Gesamtwahrscheinlichkeit $P(x_1, \ldots, x_M \mid \boldsymbol{F})$ gesucht, die allgemein durch folgende Gleichung modelliert wird:

$$P(x_1, \ldots, x_M \mid \boldsymbol{F}) = \frac{1}{\zeta} \prod_m \phi(x_m \mid \boldsymbol{F}) \prod_{m>1} \psi(x_m, x_{\pi(m)} \mid \boldsymbol{F}). \tag{2.4}$$

Dabei stellt ζ wiederum einen Normalisierungsterm dar. Die unären Potentialfunktionen ϕ geben über die Beziehung zwischen den beobachteten Daten $\boldsymbol{F}$ und der Klassenlabeln Auskunft, die binären Potentialfunktionen ψ geben die Beziehungen zwischen den Klassenlabeln benachbarter Bildregionen unter Berücksichtigung beobachteter Merkmale wieder.

Wir übertragen nun dieses Konzept auf die Notation der Bayes-Netze und führen den Begriff *Bedingtes Bayes-Netz* ein. Darunter verstehen wir ein Bayes-Netz, d. h. einen zyklenfreien Graphen mit gerichteten Kanten, dessen Knoten bedingte Zufallsvariablen repräsentieren und dessen Übergangswahrscheinlich wie die Zufallsvariablen von global beobachteten Merkmalen $\boldsymbol{F}$.

Konkret leiten wir die Struktur des Bedingten Bayes-Netzes aus der Regionenhierarchie ab. Jede Region S_m wird durch eine Zufallsvariable x_m repräsentiert, die die Klassenzugehörigkeit der Region angibt und diese als Wahrscheinlichkeiten für die K Zustände angibt. Die Kanten zwischen den Zufallsvariablen stammen aus dem Regionen-Hierarchiegraph, wobei sich die Richtung der Kanten ändert. Dies begründen wir inhaltlich damit, dass wir den Einfluss der

Klassenlabel von Objekten auf die Klassenlabel der Objektteile modellieren wollen und nicht umgekehrt. Nebenbei hat es noch den praktischen Nutzen, dass jeder Knoten im Bedingten Bayes-Netz genau einen Elternknoten hat. Somit hat das Moralisieren unseres Bayes-Netzes nur eine Auswirkung: Es fallen die Richtungen der Kanten weg.

Im Kalkül der Bedingten Zufallsfelder können wir Gesamtwahrscheinlichkeit gemäß Gl. 2.4 modellieren. Die unären Potentialfunktionen ϕ definieren wir als

$$\phi(x_m \mid \boldsymbol{F}) = P(x_m \mid \boldsymbol{F}), \tag{2.5}$$

d. h. sie geben Auskunft über die Zusammenhänge der beobachteten Daten $\boldsymbol{F}$ und den Klassenlabeln der segmentierten Regionen. Die binären Potentialfunktionen ψ spiegeln die Beziehungen zwischen den Klassenzugehörigkeiten von hierarchisch benachbarten Regionen S_m und $S_{\pi(m)}$ wieder, weshalb wir ψ durch

$$\psi(x_m, x_{\pi(m)} \mid \boldsymbol{F}) = P(x_m \mid x_{\pi(m)}, \boldsymbol{F}) \tag{2.6}$$

definieren. Somit lernen wir die Wahrscheinlichkeiten dafür, wie die Zugehörigkeit einer Region S_m zu den verschiedenen Klassen in Abhängigkeit ihres Elternknotens und der beobachteten Merkmale beider Regionen vorkommt.

Die beiden Wahrscheinlichkeiten aus den Gl. 2.5 und 2.5 leiten wir aus Trainingsdatensätzen ab. Die Wahrscheinlichkeiten $P(x_m \mid x_{\pi(m)}, \boldsymbol{F})$ können wir durch Auswerten der Klassenzugehörigkeiten in den bestimmten Regionenhierarchie-Relationen der Trainingsdaten ermitteln, und die Wahrscheinlichkeiten $P(x_m \mid \boldsymbol{F})$ bestimmen wir durch einen Klassifikator. Da wir Daten zum Trainieren eines Klassifikators haben, können wir eine überwachte Klassifikationsmethode verwenden.

Nach Umstellung des Graphischen Modells auf ein Bedingtes Markoff-Zufallsfeld postulieren wir: Die Zufallsvariable x_m hängt nur von zwei voneinander unabhängigen Zufallsvariablen ab.

2.1.4.4. Inferenz in Bayes-Netzen

Unter Inferenz versteht man das Ziehen von Schlussfolgerungen, wenn Informationen über einzelne Zufallsvariablen bekannt werden, d. h. das Bestimmen von Posterior-Wahrscheinlichkeiten für die anderen Zufallsvariablen. Bei der Auswertung von Bayes-Netzen gibt es zwei wichtige Ziele: Einerseits ist es interessant, Randverteilungen zu bestimmen, d. h. die Wahrscheinlichkeitsverteilung für einzelne Zufallvariablen x_m, andererseits die wahrscheinlichste Klasse für jede der vorhandenden Zufallsvariablen. In komplexen Graphischen Modellen mit Hierarchien und Nachbarschaften können i. d. R. nur approximierende Inferenzverfahren verwendet werden. Für einfache Graphenstrukturen wie Ketten und Bäume gibt es aber exakte Inferenz-Verfahren, wie z. B. der Summe-Produkt-Algorithmus sowie der Max-Summe-Algorithmus, die beide ausführlich von Bishop (2006) vorgestellt und diskutiert werden.

Der Summe-Produkt-Algorithmus wird zum Bestimmen von Randverteilungen eingesetzt und ist in der Formulierung von Pearl (1997) effizient bei der Bestimmung aller Randverteilungen. Somit liegen nach Abschluss der Inferenz wiederum Wahrscheinlichkeiten für jede Zufallsvariable vor. Diese ermöglichen entweder weitere probablistische Auswertungen und Analysen, z. B. eine Untersuchung bzgl. der Überzeugung des Klassifikators vom Ergebnis (Abstand zwischen der wahrscheinlichsten Klasse und der zweitwahrscheinlichsten). Ein wichtiger Schritt dieses Algorithmus ist das schrittweise Marginalisieren von Zufallsvariablen, das mittels Summieren der Wahrscheinlichkeiten über alle Zustände durchgeführt wird. Demzufolge wird eine diskrete Modellierung der Merkmale vorausgesetzt. Allerdings können auch

kontinuierliche Merkmale verarbeitet werden, wenn man die Summenformeln durch Integrale ersetzt.

Russell & Norvig (1995) setzen den Summe-Produkt-Algorithmus als funktionale Methode um, d. h. er bestimmt für genau eine Zufallsvariable x_m die Randverteilung. Somit mit diese Methode M-mal aufgerufen werden, weshalb die Implementierung durch die Mehrfachberechnungen gleicher Wahrscheinlichkeiten ineffizient wird. Pearl (1997) stellt eine effiziente Berechnung aller einzelnen Randverteilungen vor, bei der ausschließlich univariate Merkmale verwendet werden. Dabei werden zunächst alle Informationen von der Wurzel zu den Blättern propagiert, dann alle Informationen von den Blättern zur Wurzel. Nach der Verknüpfung beider Informationen je Zufallsvariable müssen die Zustandsbelegungen nur noch normiert werden. Bishop (2006) zeigt hierbei, dass der Summe-Produkt-Algorithmus sehr effizient ist, wenn die Normierung zum Schluss durchgeführt wird: Der Normierungsterm ist kostant bei allen Zufallsvariablen.

Außerdem weist Bishop (2006) darauf hin, dass der Summe-Produkt-Algorithmus nicht immer das beste Ergebnis ausgibt, d. h. die maximale Gesamtwahrscheinlichkeit kann bei anderen Belegungen der Zufallsvariablen liegen als bei den Belegungen der maximalen Randverteilungen. Dazu gibt Bishop (2006) folgendes kleines Beispiel an. Es werden zwei binäre Zufallsvariablen betrachtet, deren gemeinsame Wahrscheinlichkeit durch

x	y	$p(x,y)$
0	0	0.3
0	1	0.3
1	0	0.4
1	1	0.0

(2.7)

angegeben wird. Demnach sind $p(x=0)=0.6$ und $p(x=1)=0.4$ sowie $p(y=0)=0.7$ und $p(y=1)=0.3$. Folglich werden bei der Klassifikation nach Anwendung des Summe-Produkt-Algorithmus die beiden Belegungen $x=0$ und $y=0$ gewählt, weil sie zum Maximum der Randverteilungen führen. Dabei ergibt die Belegungskombination $x=1$ und $y=0$ eine höhere gemeinsame Wahrscheinlichkeit.

Der zweite wichtige Inferenz-Algorithmus in einfachen Graphenstrukturen ist der Max-Summe-Algorithmus. Mit diesem Verfahren werden keine Randverteilungen mehr bestimmt, sondern es wird die Konstellation an Belegungen für alle Zufallsvariablen bestimmt, mit der die maximale Gesamtwahrscheinlichkeit erzielt wird. Allerdings erhält man hier durch den verwendeten max-Operator Zustandsbelegungen sowie die maximale Gesamtwahrscheinlichkeit, aber keine Wahrscheinlichkeiten für einzelne Zufallsvariablen. Aus diesem Grund haben wir uns gegen den Einsatz dieses Verfahrens entshcieden, obwohl es immer eine optimale Klassifikation gemäß unseres Graphischen Modells erzielt.

Stattdessen haben wir den Summe-Produkt-Algorithmus von Pearl (1997) auf unser Bedingtes Bayes-Netz übertragen. Für die Evaluation und Visualisation der Bildinterpretation haben wir geben wir dann jeweils die Klassen aus, die dem wahrscheinlichsten Zustand einer Zufallsvariable entspricht, d. h. wir bestimmen die maximalen Randverteilungen.

2.1.5. Visualisierung der Bildinterpretation

Unser Konzept endet mit der Ausgabe von Klassifikationskarten, siehe Abb. 2.1, in denen jeweils alle Regionen entsprechend der Klassifikationsergebnisse hervorgehoben. Diese lassen sich auch zu einem Bild kombinieren. Dabei ist zu beachten, dass sich die Regionen überlappen

werden, da einige komplexe Objekte, andere Teile dieser Objekte darstellen. Bei der Synthese der verschiedenen Klassifikationskarten zu einem mehrfarbigen Bild, vgl. Abb. 1.1, berücksichtigen wir somit die Bestandteilshierarchie der Objekte, d. h. zuerst werden wir die größeren Objekte visualisieren und erst im Anschluss daran deren Bestandteile.

2.2. Bestimmung von Regionen für die Interpretation

In diesem Abschnitt formulieren wir die Anforderungen für unseren Ansatz einer hierarchischen Bildsegmentierung und stellen verschiedene bereits existierende Segmentierungsansätze vor. Mit einer Bewertung der bisherigen Arbeiten motivieren wir das von uns gewählte Segmentierungsverfahren.

2.2.1. Anforderungen an die hierarchische Bildsegmentierung

Die Abb. 2.5 im vorangegangenen Abschnitt zeigt eine mögliche hierarchische Bildsegmentierung. Idealerweise liefert die Bildsegmentierung einen Baum von Regionen, in dem jede Region genau ein Objekt der abgebildeten Szene zeigt, und alle Objekte durch jeweils genau eine Region dargestellt werden. Außerdem gibt die hierarchische Anordnung der Regionen die Zugehörigkeit der kleineren Objekte zu größeren Aggregaten wieder. Aus unserer Sicht ist dieses Ideal am ehesten mit perfekten Objektdetektoren und einem logikbasierten Interpretationssystem zu erreichen. Ein Ansatz, der diese Richtung aufgreift, aber nur eine verbesserte Detektion von `window`-Instanzen erzielt, wurde von Terzić *et al.* (2010) vorgeschlagen.

Weitere Objekt-Detektoren für Fenster in Einzelbildern haben u. a. Lee & Nevatia (2004), Ali *et al.* (2007), Wenzel & Förstner (2008) oder Reznik (2009) entwickelt. Instanzen der Klasse `door` werden auch bei dem auf einer Grammatik für Gebäudefassaden basierten Verfahren von Ripperda (2008) berücksichtigt. Der aufmerksamkeitsbasierte Blobdetektor von Jahangiri & Petrou (2009) liefert Blobs mittlerer Ausdehnung, die gut als Fenster, Türen oder Balkone identifiziert werden können, wenn ein entsprechendes Gebäudemodell vorliegt (Xu *et al.*, 2010). Dieser Detektor gibt das dargestellte Objekt in seiner Umgebung und somit statt der exakten Kontur ein Rechteck aus, in dessen Inneren das Objekt liegt.

Für die Segmentierung von großen Objekten in Bildern kann man die Verfahren von Kumar & Hebert (2003a), Korč & Förstner (2008) und Verbeek & Triggs (2007) integrieren, die Bilder mittels Markoff-Zufallsfeldern segmentieren. In den erwähnten Arbeiten werden in terrestrischen Fassadenaufnahmen Instanzen der Klassen `facade`, `sky` und `vegetation` bzw. `ground` gefunden. Für weitere Bestandteile von Gebäuden wie Balkone, Gauben und Treppen sind uns keine geeigneten Detektoren in Einzelbildern bekannt. Zudem wurden die oben genannten Verfahren für die Analsyse von terrestrischen Aufnahmen entwickelt und sind bislang nicht auf Luftbilder angewandt worden, mit Ausnahme von (Lee & Nevatia, 2004), siehe z. B. (Meixner & Leberl, 2010).

Da wir Instanzen von K Objekttypen in verschiedenen Aufnahmesituationen detektieren wollen, haben wir uns gegen die Modellierung vieler Objekt-Detektoren entschieden. Stattdessen schlagen wir eine Segmentierung von Regionen vor, die auf die Bestimmung einfacher Bildmerkmale, wie Blobs oder Bildsegmente beruhen. Zur Beurteilung der verschiedenen Vorgehensweisen schlagen wir folgende Kriterien vor. Die Reihenfolge haben wir absteigend nach der Bedeutung des Kriteriums festgelegt. Im anschließenden Abschnitt werden wir verschiedene Verfahren zur Segmentierung von Regionen vorstellen und sie hinsichtlich dieser Kriterien bewerten.

1. **Größe:** Es müssen Regionen in verschiedenen Bildmaßstäben segmentiert werden, um Regionen verschiedener Größen zur Beschreibung der Instanzen aller K verschiedenen Objekttypen verwenden zu können.

2. **Baum:** Die Hierarchie der detektierten Regionen soll eine Baumstruktur aufweisen. Nur bei einer vollständigen Inklusion von kleinen Region in großen Regionen ist die Zuordnung immer eindeutig.

3. **Kontur:** Um für das Objekt charakteristische Merkmale bestimmen zu können, ist die genaue Erfassung der Objektgrenzen wünschenswert. Sonst können keine formbasierten Merkmale für die Klassifikation der Regionen verwendet werden.

4. **T+L:** Das Segmentierungsverfahren muss datenbasiert arbeiten und nicht modellbasiert, damit es sowohl auf terrestrischen Aufnahmen als auch auf Luftbildern einsetzbar ist.

5. **Nachbarn:** Eine Partitionierung des Bildes erscheint vorteilhaft, um so auch Merkmale von benachbarten Regionen bei der Merkmalsextraktion bestimmen zu können.

6. **Duplikate:** Bei Segmentierungen in mehreren Bildmaßstäben kann ein Objekt auf mehrere, nur leicht unterschiedliche Regionen abgebildet werden. Diese vielfach segmentierten Instanzen eines Objekts, die Duplikate, bedeuten einen Mehraufwand bei der Auswertung.

2.2.2. Bisherige Verfahren zur Segmentierung von Regionen

Eine Alternative zu den oben erwähnten klassenspezifischen Objekt-Detektoren stellt der Ansatz von Borenstein & Ullman (2004) dar. Hier erfolgt die Einteilung des Bildes in quadratische Bildblöcke. Die Lage, Orientierung und Größe der Blöcke wird durch einen Blob-Detektor wie z. B. (Lowe, 2004) oder Matas *et al.* (2002) bestimmt, d. h. eine vollständige Überdeckung des Bildes durch Blobs ist nicht gewährleistet. Die hierarchische Struktur ist nur dort bestimmbar, wo es viele sich überlappende Blobs gibt. Eindeutigkeit beim Aufbau der hierarchischen Struktur ist nicht gegeben bzw. muss künstlich erzwungen werden, wenn sich kleine Blobs mit mehreren Großen überlappen. Das Verfahren konnte im Bezug auf Gebäude bislang nur auf stark geglätteten und sehr stark verkleinerten Fassadenbildern evaluiert werden (Lifschitz, 2005). Eine Erkennung weiterer Gebäudeteile ist dann nicht mehr möglich. Es wurde bislang nicht untersucht, ob der Ansatz von Borenstein & Ullman (2004) bei Luftbildern eingesetzt werden kann bzw. wieviele Duplikate von Objekten die Hierarchie enthält.

Im Kontext der Fassadeninterpretation haben Burochin *et al.* (2009) und Hernández & Marcotegui (2009) Verfahren zur hierarchischen Zerlegung von Fassaden in rechteckige Bildausschnitte entwickelt. Sie nutzen die vielen horizontalen und vertikalen Kanten an einer Fassade zur Einteilung des Bildes in Regionen. Allerdings wurden beide Ansätze nur auf entzerrten Fassadenbildern evaluiert, und eine Möglichkeit der Erweiterung der Ansätze auf die Segmentierung von Luftbildern sehen wir nicht, weil Dachstrukturen häufig nicht rechteckig sind. Somit erfüllen diese Ansätze vier Kriterien (Größe, Baum, Nachbarn und Duplikat).

Kumar & Hebert (2003b), Gould *et al.* (2008) und Plath *et al.* (2009) verwenden Hierarchien von Markoff-Zufallsfelder zur Segmentierung von Bildern. Ausgegeben werden semantisch sinnvolle Komponenten, die sich nicht überlagern und eine Bildpartitionierung bilden. Die Hierarchie der Regionen wird nur zur Extraktion von Merkmalen in höheren Bildmaßstäben

verwendet, um die Klassifikation zu verbessern. Das Erkennen von Objektteilen wird nicht angestrebt: das für uns sehr wichtige Kriterium bzgl. der Größe wird nicht erfüllt.

Es folgen nun einige Segmentierungsverfahren, die Regionen auf Grund eines Ähnlichkeitsmaßes oder eines Homogenitätskriteriums bilden. Hierarchische Segmentierungen erhält man entweder durch Anwendung der Verfahren in mehreren Bildmaßstäben, d. h. im Skalenraum des Bildes, oder durch Defintion einer irregulären Bildpyramide. Skalenräume finden in der Literatur seit der Arbeit von Witkin (1983) Beachtung und wurden intensiv von Koenderink (1984), Perona & Malik (1990) und (Lindeberg, 1994) untersucht. Der Gauß'sche Skalenraum, definiert durch Glättungen des Bildes mit verschieden starken Gauß-Filtern, unterbindet laut (Koenderink, 1984) willkürliche Veränderungen des Bildsignals sowie das Auftreten neuer Bildstrukturen am besten und wird daher sehr häufig verwendet.

Die Bildeinteilung in Regionen und Hintergrund von Förstner (1994) verlangt ausschließlich kleine Gradientenstärken in der Umgebung von Pixeln, damit diese zu einem homogenen Bereich zusammengefasst werden. Durch Segmentierung des Bildes in mehreren Bildmaßstäben kann eine hierarchische Anordnung der homogenen Regionen konstruiert werden, allerdings hat diesbzgl. noch niemand die Entwicklung der homogenen Regionen im Skalenraum untersucht. Somit ist eine baumförmige Struktur der Regionen nicht garantiert. Zudem zeigen die detektierten Regionen selten vollständige Objekte, da sich die Regionen nicht bis zu den Rändern erstrecken. Am Objektrand gibt es i. d. R. starke Gradienten, so dass dort Bildkanten und keine homogenen Regionen mehr erkannt werden. Eine Partitionierung des Bildes ist nicht notwendig, da Fuchs (1998) einen Nachbarschaftsgraphen zwischen den homogenen Regionen definiert hat. Bislang gibt es keine Untersuchungen, wie sich die Regionen bei der Segmentierung in mehreren Bildmaßstäben verhalten.

Drei Verfahren zur lückenlosen Einteilung des Bildes in sich nicht schneidende Regionen und deren Verhalten bei Segmentierung in mehreren Bildmaßstäben hat Olsen (1996) in Bezug auf medizinische Daten untersucht. Drauschke *et al.* (2006b) haben diese Verfahren auf Eignung zur Segmentierung von Luftbildern analysiert. Der Wasserscheidenalgorithmus schneidet in beiden Vergleichen am besten ab, wenn Regionen mit Konturen von kleinen und großen Objekten auf Basis homogener Farbwerte, d. h. der gewichteten Gradienten, gebildet werden. Hierarchien von Wasserscheidenregionen wurden durch Gauch (1999) untersucht: eine baumartige Struktur wird durch seinen Ansatz erzielt. Somit erfüllt die Segmentierung mit Wasserscheiden im Skalenraum fast alle Kriterien: Nur das Problem der mehrfachen Segmentierung von Regionen ist gegeben, wenn man sehr viele Bildmaßstäbe betrachtet.

Neben der Segmentierung in mehreren Bildmaßstäben sind auch irreguläre Pyramiden gut für die Generierung hierarchischer Bildsegmentierungen geeignet. In jeder Hierarchieebene wird das segmentierte Bild als Graph interpretiert: Die Menge der Knoten entspricht der Menge der segmentierten Regionen, die Kanten werden durch den Nachbarschaftsgraphen der Regionen definiert. Von einer Hierarchiebene zur nächsten wird der Graph um einige Elemente reduziert, d. h. einige benachbarte Regionen werden vereint und der Nachbarschaftsgraph entsprechend aktualisiert. Diese Vorgehensweise der Graphkontraktion wurde v. a. von Kropatsch (1995) diskutiert. In Abb. 2.10 illustrieren wir das Konzept von irregulären Pyramiden mit drei Hierarchieebenen. Links sind die jeweiligen Segmentierungen zu sehen, in der jeder Kreis für ein Pixel steht. Auf der rechten Seite stellen wir die entsprechenden Graphen dar. Die Zugehörigkeit der Pixel zu Regionen haben wir über die Farbgebung in den Kreisen markiert.

Meer (1989) realisiert seine stochastische, irreguläre Pyramide, indem für jede Region eine von den Nachbarregionen unabhängige Zufallsvariable definiert wird, deren Belegung über das Fortbestehen der Region entscheidet. In der Originalarbeit entscheidet ein Homoge-

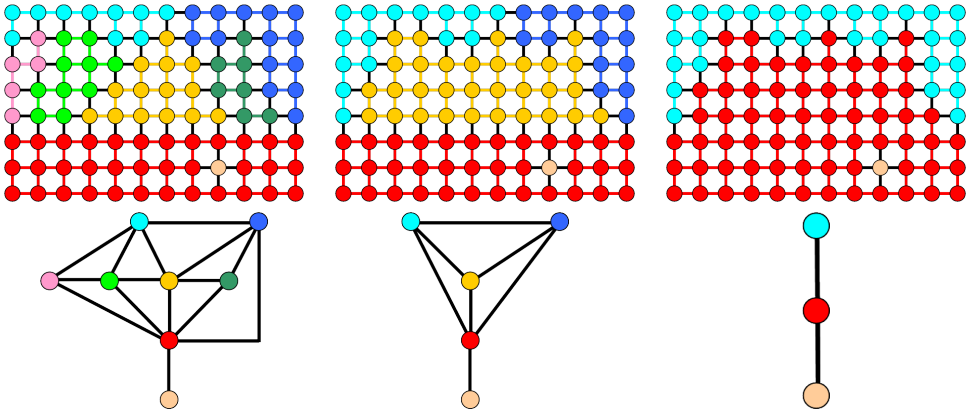


Abbildung 2.10.: Drei Hierarchiestufen einer irreguläre Bildpyramide mit den Regionen der drei Hierarchiestrufen (oben) und den entsprechenden Graphen (darunter).

nitätskriterium darüber, welche Regionen miteinander verschmelzen. Dagegen verwendet Pan (1994) einen MDL-Ansatz, um den Verschmelzungsprozess der Regionen zu steuern. Beide Verfahren wurden von Schuster (2002) implementiert und auf ihre Verwendbarkeit zur Segmentierung von Luftbildern untersucht: Beide Segmentierungen enthalten zu viele schlecht interpretierbare Regionen in den Pyramidenstufen.

(Guigues *et al.*, 2003) und (Marfil *et al.*, 2007) verwenden Ähnlichkeitsmaße zur effizienten Konstruktion von irregulären Bildpyramiden. Neben der Wahl des entsprechenden Kriteriums unterscheiden sich diese Verfahren auch darin, ob das Kriterium nur für jeweils zwei Regionen oder in einem größeren Umfeld einer Region überprüft wird. So wird der Verschmelzungsprozess von Guigues *et al.* (2003) als Cluster-Algorithmus von Sub-Graphen formuliert. Beide Verfahren sind universell einsetzbar. Ob die Wahl der jeweiligen Ähnlichkeitsmaße zu einer guten Segmentierung von Objekten und ihren Teilen in Fassadenbildern und Luftbildern führt, wurde bislang noch nicht untersucht.

Der sehr aktuelle Ansatz zu einer hierarchischen Bildsegmentierung von Arbeláez *et al.* (2011) nutzt lokale Strukturanalysen und auf Homogenität basierte Gruppierungen von Pixeln zu Regionen zur Erstellung eines hierarchischen Baums von Regionen. Wegen der Berücksichtigung vielseitiger lokaler und globaler Informationen erzielt diese Arbeit vielversprechende Ergebnisse für die hierarchsiche Bildinterpretation. Wir konnten diesen Ansatz in unserer Arbeit aber nicht mehr berücksichtigen.

Wir fassen die betrachteten Ansätze zur Segmentierung von Regionen und ihr Verhalten in Bezug auf die von uns genannten Kriterien in Tab. 2.1 zusammen. Jedes in der Tabelle erwähnte Verfahren steht dabei für die Gruppe von Segmentierungsverfahren, die wir im entsprechenden Abschnitt betrachtet haben.

Der Ansatz von Arbeláez *et al.* (2011) könnte der beste für unsere hierarchische Segmentierung sein, allerdings ist er erst kürzlich publiziert worden. So konnten wir ihn nicht mehr bzgl. der Segmentierung von Gebäudeaufnahmen untersuchen. Da er aber auch Wasserscheidenregionen verwendet, stärkt das auch unsere Wahl für dieses Segmentierungsverfahren: Da die Wasserscheidensegmentierung im Skalenbaum am besten abschneidet, verwenden wir diesen Ansatz stellen wir es im folgenden Abschnitt etwas ausführlicher vor.

Tabelle 2.1.: Verhalten der Segmentierungsalgorithmen. Die Kriterien wurden durchnummiert, entsprechend der Auflistung oben. Ein Strich markiert, dass das Kriterium nicht erfüllt wird, ein + gibt ein erfülltes Kriterium an. Bei ? wurde das entsprechende Verfahren noch nicht in Bezug auf das Kriterium untersucht.

Verfahren	Größe	Baum	Kontur	T+L	Nachbarn	Duplikat
Borenstein (2004)	-	-	-	?	-	+
Hernández (2009)	+	-	+	-	+	+
Plath (2009)	-	+	+	+	+	?
Förstner (1994)	+	?	-	+	+	?
Gauch (1999)	+	+	+	+	+	?
Meer (1989)	+	+	-	+	+	?
Marfil (2007)	+	+	?	+	+	?
Arbeláez (2011)	+	+	+	+	+	+

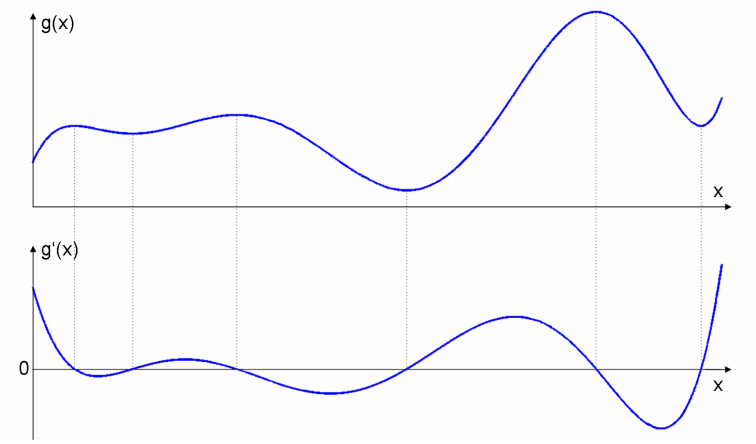


Abbildung 2.11.: Prinzip des Wasserscheidenalgorithmus

2.2.3. Wasserscheidenalgorithmus

Das Konzept von der Einteilung eines Geländes mittels Wasserscheiden wurde erstmals von Maxwell (1870) vorgeschlagen. Das Gelände wird durch von unten gleichmäßig aufsteigendes Wasser geflutet. Das Wasser wird also in den lokalen Minima des Geländes als erstes sichtbar. Auf den Geländekämmen rund um jedes lokale Minimum treffen die Flutungsbecken aufeinander, dort werden die Regionengrenzen abgesteckt. In Abb. 2.11 zeigen wir ein eindimensionales Signal g und seine erste Ableitung, zudem haben wir die Extremstellen markiert.

Algorithmisch können die Wasserscheidenregionen auf zwei Arten berechnet werden. Entweder beginnt die Regionensuche in den bestimmten lokalen Minimalstellen und die Konturen der Regionen werden durch ein simultanes Regionwachstum bestimmt. Oder man rutscht von allen Punkten des Geländes bergab zu den Minimalstellen und bestimmt so die Zugehörigkeit zu Regionen. Einen Vergleich beider Techniken liefert die Arbeit von Lin *et al.* (2006).

Üblicherweise wird der Wasserscheidenalgorithmus nicht auf den Intensitätswerten des Bildes, sondern auf einem Gradientenstärkebild angewendet, siehe (Vincent & Soille, 1991). Wenn man das Gradientenstärkebild als 3D-Gelände darstellt, werden die Regionengrenzen dort gezogen, wo die Intensitätsunterschiede zwischen benachbarten Pixeln am größten sind. Wasserscheidenalgorithmen, die direkt auf den Intensitäten des Bildes arbeiten, werden eher zur

Bildverbesserung als zur Segmentierung verwendet, siehe (Fairfield, 1992).

Implementierungen des Wasserscheidenalgorithmus mit subpixelgenauen Regionengrenzen wurden bereits von Steger (1999) und Meine & Köthe (2006) vorgestellt. Damit ist das Verfahren auch für kontinuierliche Bilddaten verwendbar bzw. kann eingesetzt werden, wenn die ausgegebenen Regionen nicht durch eine Agglomeration von Pixeln, sondern durch kontinuierliche Flächen repräsentiert werden sollen.

2.3. Klassifikation der Regionen als Basis für die Interpretation

In diesem Abschnitt formulieren wir unsere Anforderungen an die Klassifikation der Regionen auf Basis der extrahierten Merkmale. Die Abhängigkeit von extrahierten Merkmalen wird durch die Wahrscheinlichkeit $P(x_m \mid \boldsymbol{F})$ modelliert, welche durch einen Klassifikator bestimmt wird. Nun motivieren wir in diesem Abschnitt die Wahl unseres Klassifikators.

2.3.1. Merkmale von Regionen

In der Literatur können wir zahlreiche Vorschläge für die Bestimmung von Merkmalen finden. Ballard & Brown (1982) haben bereits vor fast dreißig Jahren eine Reihe von guten Merkmalen zur Erkennung von Objekten anhand von Form, Textur oder Farbe angegeben. Neben diesen mittlerweile klassischen Merkmalen gibt es noch viele andere, neuere Merkmale z. B. für die Beschreibung von Texturen (Shotton *et al.*, 2006) oder für die Analyse von Konturen (Gu *et al.*, 2009). Die Deskriptoren von Lowe (2004) oder Dalal & Triggs (2005) können auch für die Klassifikation von gleichförmigen Bildbereichen verwendet werden, eine Adaption für die Merkmalsbeschreibung von Regionen halten wir möglich.

Guyon & Elisseeff (2003) empfehlen, möglichst viele Merkmale zu erzeugen und diese gegebenfalls noch miteinander zu kombinieren, um noch bessere Klassifikationsergebnisse zu erzielen. Dann könnte man bei unbegrenzter Rechenleistung und Rechenzeit alle bisher vorgeschlagenen Merkmale aufgreifen und anschließend noch die Anzahl der Merkmale beliebig vergrößern. Aus Komplexitätsgründen werden wir uns aber auf wichtige Merkmale beschränken, deren Anzahl wir mit D bezeichnen.

2.3.2. Anforderungen an die Klassifikation der segmentierten Regionen

Die Bildsegmentierung liefert M Regionen unterschiedlicher Größe, die K verschiedene Objekte repräsentieren. Wir sind davon überzeugt, dass viele verschiedene Merkmale notwendig sind, um die verschiedenen Klassen im Bereich der Gebäudeerkennung voneinander trennen zu können. Formmerkmale sind hilfreich, um die rechteckigen Strukturen von Fassade, Fenster und Türen von komplizierteren Formen bei Vegetation oder Dachaufbauten zu unterscheiden. Ebenso hilfreich können Farb- oder Texturmerkmale sein. Schließlich können wir so einen blauen Himmel mit weißen Haufenwolken von grün belaubten Bäumen mit braunem Stamm oder einer roten Ziegelwand einer Fassade unterscheiden. Die Anzahl der Merkmale bezeichnen wir mit D.

Die Variabilität von Gebäuden und ihren Bestandteilen ist sehr groß. Sie unterscheiden sich u. a. auf Grund verschiedener Materialien, Größen, Formen und Farben. Würden wir einen Klassifikator auf nur einem modernen Innenstadtbau mit einer schlichten Fassade und einfachen Fenstern trainieren, so können wir nicht erwarten, dass derselbe Klassifkator in Einfamilienhäusern oder reichlich verzierten Barockbauten mit Balkonen und vielfach unterteilten Fenstern erfolgreich klassifizieren kann. Aus diesem Grund haben wir den Ansatz von

Fei-Fei *et al.* (2006) nicht in Erwägung gezogen, die beim Lernen möglichst wenige Beispiele verwenden.

Unüberwachte Klassifikatoren fügen diverse Objekte nach der Ähnlichkeit der Merkmale zu Gruppen zusammen oder teilen sie durch die Differenz in den Merkmalen in unterschiedliche Gruppen ein. Da wir umfangreiche Testdatensätze mit manuell erstellten Annotationen verwenden können, haben wir die Möglichkeit, den Erfolg eines Klassifikators direkt zu bewerten. Daher haben wir bei der Wahl einer geigneten Klassifikationsmethode auf überwachte Klassifikationen geachtet. Folgende Kriterien haben wir bei der Auswahl überprüft:

1. **Klassenanzahl:** Der Klassifikator muss K Klassen unterscheiden können, deren Auftreten ungleich verteilt ist. Ein Verfahren das mit extrem kleinen und extrem großen Klassengrößen gleichzeitig umgehen kann, wäre wünschenswert. Wenn aber a-priori Wahrscheinlichkeiten für das Auftreten einer Klasse zwischen 0.5% und 30% schwanken kann, dann können die kleinen Klassen auch erstmal zusammengefasst werden, anschließend wird die Klasifikation auf diesen erneut durchgeführt. Wichtig an dieser Stelle ist daher vor allem: Der Klassifikator muss $K > 2$ Klassen unterscheiden können.

2. **Ausgabe:** Der Klassifikator muss Wahrscheinlichkeiten ausgeben, die die Zugehörigkeiten zu den verschiedenen Klassen angeben.

3. **D+K:** Der Klassifikator muss sowohl diskrete als auch kontinuierliche Merkmale verwenden können. Eine Disketisierung kontinuierlicher Merkmale ist hierbei möglich, um einheitlich geartete Merkmale und deren Verarbeitung zu gewährleisten.

4. **Verteilungen:** Der Klassifikator muss mit beliebigen Verteilungen der Merkmale umgehen können: Viele der extrahierten Merkmale sind nicht normalverteilt.

5. **Datenmenge:** Der Klassifikator muss effizient sein: Wir wollen mit möglichst vielen unserer M Datensätze trainieren, 80% der Daten zum Trainieren verwenden. Das bedeutet, dass bei einem Datensatz von 100 Bildern, in denen 1000 Regionen je Bild segmentiert werden, ca. 80 000 Daten zum Lernen verwenden werden können. Ein zweiter Grund für die Effizienz ist die Anzahl der Merkmale: $D > 50$.

6. **Selektion:** Wenn der Klassifikator Aussagen über die Bedeutung der verwendeten Merkmale machen kann, können diese für die Modellierung der Wahrscheinlichkeiten zwischen $\underline{x}_m$ und seinem Elternteil $\underline{x}_{\pi_m}$ weiter verwendet werden.

Im Folgenden stellen wir einige Standard-Klassifkationsverfahren vor, die bei vielen aktuellen Problemstellungen eingesetzt werden. Die theoretischen Grundlagen zu den jeweiligen Verfahren werden von Duda *et al.* (2001) und Bishop (2006) beschrieben und diskutiert. Eine vorangehende Merkmalsreduktion mittels Hauptkomponentenanalyse (PCA) kommt nicht in Frage, da die PCA eine Projektion in den Unterraum darstellt, der die gegebenen Daten am besten beschreibt. Da Fassaden, Fenster und Türen rechteckig sind, erwarten wir, dass formbeschreibende Merkmale am Aufspannen des Unterraum beteiligt sind. Folglich könnten wir einige Klassen nicht mehr von einander unterscheiden.

Zunächst möchten wir kurz auf die Bestimmung der Maximalen-Likelihood-Wahrscheinlichkeit (ML) bzw. der Maximalen a-posteriori-Wahrscheinlichkeit (MAP) unter Verwendung von Mischverteilungen eingehen, die man gut mit dem EM-Algorithmus bestimmen kann. Zur Berechnung einer guten Approximation der Mischverteilung durch mehrere Normalverteilungen müssen auch die Kovarianzen zwischen den Merkmalen berechnet werden. Da wir sowohl diskrete als auch kontinuierliche Merkmale haben, können wir diese Berechnungen nur

durchführen, wenn wir die Merkmale einheitlich modellieren. Eine Modellierung der diskreten Zufallsvariablen als kontinuierliche Größen lehnen wir ab, da so der Wertebereich in undefinierte Bereiche erweitert wird (z. B. durch Zulassen negativer Werte). Für die Bestimmung diskreter Wahrscheinlichkeiten im D-dimensionalen Raum ist unser D zu groß und die Anzahl der Regionen M pro Bild zu klein, um numerisch stabile Ergebnisse zu erzielen.

Die Lineare Diskriminanzanalyse (LDA) projiziert die Daten in den Unterraum, der die Klassen am besten trennt, d. h. es wird ein Unterraum gesucht, in dem die Varianz zwischen den Klassen maximal und die Varianzen innerhalb der Klassen minimal sind. Bei der LDA werden nun lineare Trennfunktionen wie Hyperebenen verwendet, um die Klassenzugehörigkeit zu bestimmen. Roth & Steinhage (1999) haben dieses Verfahren um Kernel-Funktionen erweitert: Dazu werden die Daten mit Kernel-Funktionen in einen höher-dimensionalen Raum projiziert, in dem sie linear trennbar sind. Auf diese Weise können nichtlineare Trennfunktionen im originalen Merkmalsraum effizient realisiert werden. In hoch-dimensionalen Räumen ist die LDA numerisch instabil, da sie auf einer Eigenwertzerlegung basiert. Aus diesem Grund wollen wir die LDA nicht generell verwenden, werden sie aber zur Bestimmung einer Referenzklassifikation nutzen.

Die Supportvektormaschinen (SVM) suchen die Trennfunktion zwischen den Klassen, die den größten Abstand zu den nächstgelegenen Daten hat. Boyd & Vandenberghe (2008) zeigen, dass diese Suche durch ein konvexes Optimierungsproblem gelöst werden kann. Dadurch liefern SVMs sehr gute Klassifikationsergebnisse. Da wir aber sehr viele Trainingsdaten verwenden wollen und die Klassen sich auch teilweise stark überlappen, sind SVMs ziemlich langsam. Wir haben unsere Experimente nach mehreren Tagen abgebrochen, da uns das Training mit SVMs nicht effizient genug ist.

Entscheidungsbäume bieten eine hierarchische Möglichkeit, Daten zu klassifizieren. Erst wird nach einem Merkmal eine Trennung der Daten erzielt, dann folgt eine weitere Teilung durch eine neue Entscheidung. Bei deterministischen Entscheidungsbäumen versucht man, die besten Merkmale für die Entscheidungen zu finden. Wenn die Bäume zu tief modelliert werden, dann passen sich die Entscheidungsbäume zu stark an die Daten an (Overfitting). Eine zufallsbasierte Variante mit vielen Entscheidungsbäumen hat Breiman (2001) vorgeschlagen. Hier werden alle Entscheidungen (Merkmal sowie Schwellwert) per Zufall getroffen, in den Blättern der Bäume befinden sich dann Wahrscheinlichkeiten zur Klassenzugehörigkeit, die über alle Bäume kumuliert werden. Eine deutliche Verbesserung erhält man, wenn man den Auswahlprozess evaluiert, d. h. unter mehreren zufällig bestimmten Möglichkeiten die beste wählt. Ein gutes Auswahlkriterium für gut balancierte Entscheidungen bzgl. der Klassifikation geben Geurts *et al.* (2006) an.

Neben den Entscheidungsbäumen gibt es noch ein weiteres Verfahren zur Klassifikation von Daten, das wir für sehr gut geeignet halten, mehrere Klassen voneinander zu unterscheiden. Schapire (1990) schlägt vor, dass es durchaus besser sein kann, bei der Klassifikation mehreren nicht ganz so optimale Klassifikatoren zu vertrauen, als sehr viel Energie in die Modellierung eines einzigen fast perfekten Klassifikators zu investieren. Im folgenden Unterabschnitt stellen wir das AdaBoost-Verfahren näher vor.

2.3.3. AdaBoost: Adaptives Boosting

Adaptive boosting (AdaBoost) ist seit seiner Optimierung durch Schapire & Singer (1999) ein sehr erfolgreiches Werkzeug für die binäre Klassifikation. Der Grundgedanke, mehrere gute Klassifikatoren zu kombinieren, wurde in vielen Applikationen um zahlreiche Ansätze erweitert. Die bekannteste AdaBoost-Anwendung ist unserer Meinung nach die Gesichtser-

kennung von Viola & Jones (2001b), die dafür eine kaskadenförmige Anordnung der einzelnen Klassifikatoren wählten.

Die generelle AdaBoost-Vorgehensweise ist die Folgende: Wir weisen allen Daten ein Gewicht zu, üblichwerweise startet man mit einer Gleichverteilung der Daten. Dann wird der erste Klassifikator κ_1 bestimmt, wobei es nur eine Anforderung an die schwachen Klassifikatoren gibt: Sie müssen jeweils besser klassifizieren als der Zufall. Nach der Evaluation des Klassifikators werden die Daten neu gewichtet: Der nächste Klassifikator soll sich dabei auf die falsch klassifizierten Daten fokussieren, deren Gewichte hoch gesetzt werden. Dieser Ablauf wird so lange durchgeführt, bis alle T Klassifikatoren κ_t bestimmt wurden. Der kombinierte Klassifikator κ ergibt sich dann aus

$$\kappa(f_m) = \operatorname{sgn} \sum_{t=1}^{T} \alpha_t \kappa_t(f_m), \tag{2.8}$$

wobei f_m der Merkmalsvektor der Region S_m ist und α_t das Gewicht des t-ten Klassifikators ist, das dessen Klassifikationserfolg wiedergibt. Bei dieser Berechnung wird ausgenutzt, dass die einfachen Klassifikatoren binär klassifizieren, d. h. entweder $+1$ oder -1 ausgeben. Ist die Summe der Gewichte der positiv antwortenden Klassifikatoren größer als die Summe der Gewichte der negativ antwortenden Klassifikatoren, dann ist die Ausgabe des kombinierten Klassifikators $+1$ bzw. anderenfalls -1.

Der Erfolg von AdaBoost beruht auf der Kombination mehrerer einfacher und sehr effizienter Klassifikatoren, deren Auswahl und Gewichtung für die binäre Klassifikation optimiert worden ist. Für viele Anwendungen wie Schrifterkennung oder Klassifikation von Regionen in Bildern sind binäre Klassifikationen unzulänglich. Eine Erweiterung des AdaBoost-Konzepts auf den Mehrklassen-Fall ist nicht so einfach. Freund & Schapire (1996) haben gezeigt, dass das Kriterium für die einzelnen Klassifikatoren κ_t, besser zu sein als der Zufall, im Mehrklassen-Fall zu schwach ist. Andererseits ist aber auch das Kriterium aus dem binären Fall (besser als 50% zu sein) zu schwer, da das im Mehrklassen-Fall nur selten eingehalten werden kann.

So führen Schapire & Singer (1999) und Allwein *et al.* (2000) die Klassifikation mit mehreren Klassen auf eine binäre Klassifikation zurück. Dazu gibt es zwei Möglichkeiten: Entweder man manipuliert den Merkmalsraum, oder man manipuliert die Klassen. Manipulation des Merkmalsraums bedeutet, dass aus dem Datensatz $(f_m, \widetilde{x}_m)$ bestehend aus Merkmalen und Klassenlabel K neue Datensätze erstellt werden, deren Merkmalsraum um eine Dimension erweitert wurde. Das neugeschaffene Merkmal enthält je eines der möglichen Klassenlabel, die von 1 bis K durchnummeriert werden. Nur der neue Datensatz mit dem Merkmalsvektor $[f_m, \widetilde{x}_m]$ erhält das neue Klassenlabel 1 (positive Klasse), die anderen $K-1$ Datensätze erhalten das neue Klassenlabel -1 (negative Klasse). Bei diesem Verfahren wird nicht nur die Dimension des Merkmalsraums vergrößert, sondern auch die Anzahl der Datensätze. Zudem variieren die relativen Häufigkeiten der beiden so entstandenen Klassen stark.

Alternativ kann man auch die Klassen neu definieren, d. h. man klassifiziert eine Klasse gegen alle anderen, die zusammen eine neue Rest-Klasse bilden, bzw. jede Klasse gegen jede andere. Die letzte von beiden genannten Varianten führt zu $\binom{K}{2}$ binären Klassifikationen, deren Berechnungen signifikant mehr kosten. Aber auch bei einer qualitativen Einschätzung dieser Herangehensweisen finden wir sie nicht gut, da nicht vorgesehene Klassifikationsergebnisse auftreten können. So kann ein Datensatz in allen K Fällen als Rest-Klasse klassifiziert werden, oder es gibt bei mindestens zwei Klassen ein positives Resultat. In der Variante, wo einzelne Klassen gegen einander klassifiziert werden, gibt es zudem die Möglichkeit, dass bei der Klassifikation zwischen den Klassen 1 und 2 die Klasse 2 gewinnt, zwischen den Klassen 2

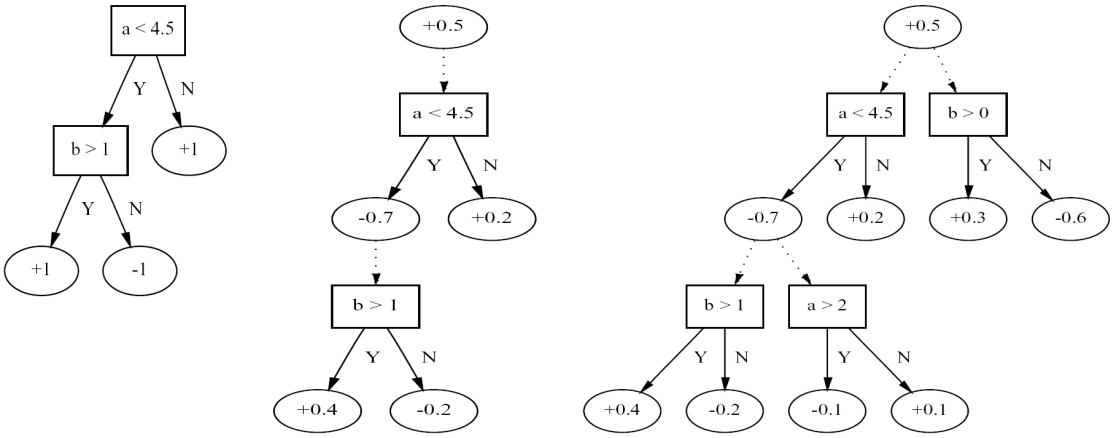


Abbildung 2.12.: Entscheidungsbaum und Alternierender Entscheidungsbaum im Vergleich (aus (Freund & Mason, 1999)).

und 3 die Klasse 3 gewinnt und zwischen den Klassen 1 und 3 die Klasse 1 gewinnt. Um diese Sonderfälle zu verhindern, bevorzugen wir eine echte Mehrklassenklassifikation.

Rätsch (2003) und Zhu *et al.* (2006) haben vielversprechende echte Mehrklassen-AdaBoost-Verfahren entwickelt. Die Klassenlabels werden von 1 bis K durchnummeriert. Die in den T Iterationen bestimmten Klassifikatoren κ_t werden dann wie folgt zum starken Klassifikator κ zusammengesetzt:

$$\kappa(f_m) = \arg\max_k \sum_{t=1}^{T} \alpha_t \mathbb{I}_{\mathrm{k}}(\kappa_t(f_m)), \tag{2.9}$$

wobei $\mathbb{I}_{\mathrm{k}}$ eine Indikatorfunktion darstellt, die 1 zurückgibt, wenn der Klassifikator κ_t eine Zugehörigkeit zur Klasse k bestimmt, und sonst 0. Die oben erwähnten Verfahren unterscheiden sich dabei in der Wahl der einfachen Klassifkatoren κ_t und ihrer Bewertung.

2.3.4. Alternierende Entscheidungsbäume

Alternierende Entscheidungsbäume (ADTboost) verknüpfen die Heransgehensweise von AdaBoost mit Entscheidungsbäumen, indem die ausgewählten Klassifikatoren κ_t hierarchisch angeordnet werden. Freund & Mason (1999) haben die binäre Variante des Verfahrens vorgeschlagen, De Comité *et al.* (2001) haben das Konzept handwerklich verbessert.

Abb. 2.12 zeigt nebeneinander v. l. n. r. erst einen Entscheidungsbaum mit zwei Entscheidungen, dann einen Alternierenden Entscheidungsbaum mit denselben beiden Entscheidungen sowie einen allgemeinen Alternierenden Entscheidungsbaum. In beiden Baumstrukturen kommen zwei verschiedene Knoten vor. Bei normalen Entscheidungsbäumen sind alle Blätter ellipsenförmig gezeichnet und stellen Knoten dar, in denen die Zugehörigkeit einer Klasse angegeben wird. Alle inneren Knoten sind dagegen rechteckig und stellen Entscheidungen bzgl. bestimmer Merkmale dar. Bei den Alternierenden Entscheidungsbäumen wechseln sich beide Knotenarten ab, wenn man einen Pfad von der Wurzel zu einem Blatt verfolgt.

Die rechteckigen Knoten haben bei Freund & Mason (1999) eine doppelte Funktion. Wir interpretieren sie einerseits als Klassifikatoren, die auf einem Merkmal eine Entscheidung bzgl. eines Schwellwerts bewirken, andererseits sind sie aber auch Vorbedingungen für die in der Hierarchie tiefer liegenden Entscheidungen. In den ellipsenförmigen Knoten stehen Werte, die

die Aussagekraft der Entscheidung nach deren Ausgang differenziert wiedergeben. Sie gehen als Gewichte α_t mit den Hoch-Indices $^+$ bzw. $^-$ bei der Kombination der Klassifikatoren κ_t ein. Im Gegenzug zu herkömmlichen Entscheidungsbäumen können sich bei ADTboost mehrere Entscheidungsknoten unterhalb eines Klassifikationsknotens befinden.

Auch der oberste Knoten eines Alternierenden Entscheidungsbaums enthält einen Wert α_0, der bei der Gewichtung der einzelnen Klassifikatoren ebenso berücksichtigt wird. Der Wert wird aus den absoluten Häufigkeiten der Klassen bestimmt und sein Wesen entspricht einem a-priori-Gewicht. Zusammen mit der hierarchischen Struktur der einzelnen Klassifikatoren und der Aufteilung der Gewichte nach Klassen haben wir drei wichtige Verbesserungen von ADTboost gegenüber AdaBoost. Dem gegenüber steht eine höhere Laufzeit von ADTboost, da bei jeder Iteration der neue beste Klassifikator unter Berücksichtigung aller möglichen Kandidaten und aller bisherigen Vorbedingungen bestimmmt wird. Bei AdaBoost werden nur die möglichen Kandidaten untersucht.

Zu den einfachen Klassifikatoren κ_t wird in den Konzepten von AdaBoost und ADTboost nichts vorgegeben, außer die Bedingung, dass sie mehr als 50% Erfolg bei der binären Klassifikation erzielen müssen. Wir werden ML-Klassifikatoren auf einzelnen Merkmalen verwenden, um einerseits die Komplexität der Suche nach dem besten Klassifikator in jeder Iteration zu beschränken, und um andererseits den Klassifikationserfolg dieser Methode auszunutzen. Auf diese Weise vereinen wir in ADTboost die drei Methoden Adaboost, Entscheidungsbäume und ML.

Wir werden ADTboost zu einer Klassifikationsmethode mit mehreren Klassen erweitern. Erste Versuche haben dazu schon Holmes *et al.* (2002) unternommen, allerdings manipulierten sie analog zu Schapire & Singer (1999) die verwendeten Klassen. Bei unserer Verallgemeinerung auf die Klassifikation von K Klassen konzentrieren wir uns vor allem auf die Auswahl der verschiedenen Hypothesen. Dazu verwenden wir aus Gründen der Effizienz relativ einfache Klassfikatoren. In zukünftigen Arbeiten können diese dann durch leistungsfähigere Klassifikatoren ersetzt werden. Die Beschränkung auf eindimensionale Merkmalsräume für die einfachen Klassifikatoren kann dann auch durch zusätzlichen Einsatz von Heuristiken bei der Auswahl von Hypothesen aufgebrochen werden, vgl. (Pfahringer *et al.*, 2001).

2.3.5. Bestimmung der Wahrscheinlichkeiten

Zu Beginn dieses Abschnitts haben wir behauptet, dass wir den Klassifikator zur Bestimmung der Wahrscheinlichkeiten $P(\underline{x}_m \mid f_m, \boldsymbol{\psi})$ nutzen, die die Abhängigkeiten der Klasse $\underline{x}_m$ von den Merkmalen f_m der segmentierten Regionen S_m angeben. Auf die bei ADTboost verwendeten Parameter $\boldsymbol{\psi}$ gehen wir in Kap. 4 ein, wenn wir die Realisierung des Klassifikators besprechen.

Die Ausgabe eines Klassifikators für K Klassen ist entweder eine dieser Klassen, d. h. aus $\{1, \ldots, K\}$ oder 0, wenn sich z. B. der Klassifikator nicht entscheiden kann oder wenn die Vorbedingung des Klassifikators durch f_m nicht erfüllt wird. Für unser Bayes-Netz benötigen wir aber Wahrscheinlichkeiten als Ausgabe der Klassifikation, die wir aus den Gewichten α_t der einfachen Klassifikatoren κ_t ableiten. Die Zugehörigkeit zur Klasse k bestimmen wir durch Auswertung aller Klassifikatoren κ_t unter Berücksichtigung der jeweiligen Vorbedingung. Dabei gibt

$$\rho_k = \alpha_0(k) + \sum_{t=1}^{T} \alpha_t \text{ mit } \kappa_t(f_m) = k \tag{2.10}$$

die Summe aller Gewichte von Klassifikatoren κ_t an, die die Daten f_m zur Klasse k zählen.

Die ρ^k sind Summen von ausschließlich positiven Zahlen. Die a-priori-Gewichtung α_0 ist positiv und sorgt somit für positive Summen. Die anderen Summanden sind nicht negativ,

und sie sind genau dann 0, wenn bei keinem Klassifikator κ_t die Vorbedingung erfüllt ist. Die Verhältniszahl

$$\frac{\rho_k}{\sum_i \rho_i} \tag{2.11}$$

gibt somit die Wahrscheinlichkeit an, wie zuversichtlich der kombinierte Klassifikator κ bei seiner Entscheidung für die Klasse k ist.

2.4. Evaluation des Konzepts

In den vergangenen drei Abschnitten dieses Kapitels haben wir unser Kozept für eine hierarchische Bildinterpretation mittels eines Bayes-Netzes vorgestellt und haben dabei auch unsere Wahl für eine bestimmte Segmentierung und eine Klassifikation der segmentierten Regionen begründet. Genau diese drei Komponenten unseres Konzepts müssen auch evaluiert werden.

Die erste Bewertung bezieht sich auf die hierarchische Segmentierung. Dabei werden wir überprüfen, inwieweit die automatisch segmentierten Regionen zu den manuell erstellten Annotationen passen. Das umfasst einerseits die flächenhafte Ausprägung der Regionen im Vergleich zu den Polygonen der Annotationen und andererseits die Hierarchie der Regionen im Vergleich zu den Teil-Beziehungen in den Annotationen.

Die zweite Evaluation betrifft den Klassifikator. Wir bestimmen Merkmale für jede segmentierte Region und ein Klassifikationstarget. Im Vergleich zu einer einfachen Klassifikation mittels LDA werden wir die Performanz des von uns implementierten ADTboost-Algorithmus untersuchen. Die Ergebnisse präsentieren wir mittels Erfolgsraten und mit Konfusionsmatrizen. Die Erfolgsraten geben den Erfolg je Klasse bzw. über alle Regionen berechnet an. Die Fehlklassifikationen zwischen den Klassen werden wir zudem in Konfusionsmatrizen darstellen.

Die dritte Evaluation bewertet die Klassifikation durch das Bayes-Netz. Durch die Hinzunahme der Information aus der Hierarchie erwarten wir eine Verbesserung der Klassifikation der einzelnen Regionen. Somit gibt es auch in dieser Bewertung eine Analyse auf der Regionenebene. Das Programm gibt aber letztendlich Klassifikationsbilder aus, in denen die entsprechend klassifizierten Pixel markiert sind. Diese ermöglichen eine zusätzliche Evaluation auf der Pixelebene.

Neben diesen quantitativen Analysen fertigen wir auch eine visuelle Ausgabe der der Bildinterpretation an, die wir zur visuellen Inspektion der erzielten Interpretationsergebnisse verwenden. Eine Herausforderung besteht hierbei in der starken Überlappung der segmentierten Regionen, da wir diese in verschiedenen Bildauflösungen finden. Auf ähnliche Weise können auch die Annotationen visualisiert werden. Damit ist ein direkter, qualitativer Vergleich zwischen mehreren symbolischen, manuell und automatisch erstellten Bildbeschreibungen möglich. Auf einen darauf aufbauenden Vergleich auf Pixelebene haben wir nicht mehr realisieren können.

Eine abschließende Evaluation ist auch auf Pixelebene möglich. Wenn wir die verschiedenen Klassifikationsbilder zu einer semantischen Bildbeschreibung zusammenfügen, dann können wir diese auch mit den zu Grunde liegenden Annotationen vergleichen.

Unsere Evaluation führen wir getrennt nach den verwendeten Datensätzen durch. In jedem Datensatz haben wir 60 bis 225 Bilder zusammengestellt, in denen zusammen mehrere tausend Objekte annotiert wurden. Bei der Segmentierung erzielen wir pro Bild etwas mehr als 1000 stabile Regionen, die wir in der Regionenhierarchie anordnen und nach Extraktion der Merkmale klassifizieren. Den Umfang unserer Stichprobe für die Evaluation unseres Konzepts halten wir daher für ausreichend groß. Alle stabilen Regionen werden bei der Überprüfung

der hierarchischen Regionenstruktur in der Evaluation berücksicht. Durch Anwendung einer Kreuzvalidierung nutzen wir auch alle Regionen bei der Evaluation der Klassifikation durch den Klassifikator bzw. durch das Bayes-Netz.

Wir verwenden vier verschiedene Datensätze für die Bewertung unseres Konzepts. Die ersten Beiden zeigen perspektivisch verzerrte Fassadenbilder und wurden von Drauschke (2009) und Korč & Förstner (2009) zusammengestellt. Sie unterscheiden sich nicht nur in der Anzahl der Auswahl der Bilder, sondern auch in der Anzahl der Klassen, deren Instanzen in den Bildern annotiert wurden. Der dritte Datensatz zeigt entzerrte Fassadenbilder und der vierte enthält Luftbildausschnitte mit Dächern von Häudern, in denen die Fassaden nur wenig sichtbar sind. Im folgenden Abschnitt werden wir diese Datensätze, d. h. die Bilder, aber auch die Annotationen und ihre jeweiligen Besonderheiten genauer vorstellen.

2.5. Verwendete Datensätze

Die ersten drei Datensätze basieren alle auf Bildern und Annotationen, die im Rahmen des Projekts *e-Training for Interpreting Images of Man-Made Scenes* (eTRIMS) aufgenommen und erstellt wurden. Wir verwenden jeweils eine Teilmenge des großen Datensatzes, sowohl in Hinblick auf die Anzahl der Bilder als auch auf die Anzahl der annotierten Klassen. Der letzte Datensatz entstand im Rahmen des Projekts *Ontologische Skalen für die automatische Erfassung, die effiziente Verarbeitung und die schnelle Visualisierung von Landschaftsmodellen.*

Ontologien und Annotationen

Korč & Schneider (2007) haben ein Werkzeug zum Annotieren von Bildern entwickelt, das in beiden eben erwähnten Projekten zum Einsatz kam. Da die Annotationen von mehreren Operatorinnen und Operatoren erhoben wurden, war eine Absprache notwendig, welche Objekte und wie sie annotiert werden sollen. Die Liste der annotierten Objekttypen findet sich nun in der Liste der verwendeten Klassen wieder, wobei bei zwei Datensätzen eine starke Vereinheitlichung der Annotationen und eine deutliche Reduzierung der annotierten Klassen vorgenommen wurde.

Das Markieren von Objekten mit Hilfe der Software von Korč & Schneider (2007) geschieht durch Umranden des Objekts mit einem einfachen Polygon. Haben Objekte Löcher, bei Luftbildern von Innenstädten kommt das z. B. bei Häusern mit eingeschlossenem Hof vor, dann bleiben diese Löcher als Bestandteil des markierten Objekts bestehen. Zudem haben wir uns in den oben genannten Projekten darauf verständigt, die Objekte vollständig zu annotieren. D. h. wird ein Gebäudeteil von einem Auto oder einem Baum verdeckt, wird der zu annotierende Rand dort markiert, wo der Operator bzw. die Operatorin ihn hinter der Verdeckkung vermutet.

Abb. 2.13 zeigt zwei annotierte Bilder. Das linke Bild zeigt ein entzerrtes Fassadenbild des dritten Datensatzes. In dem Bild wurden vier Balkone markiert. Die annotierten Vierecke der vier Balkone (türkis) zeigen allerdings mehr vom Hintergrund, d. h. den Türen, da die Geländer aus Metallstangen bestehen. Das Polygon des Bodens (magenta) wiederum zeigt mehr vom Sockel des Zauns als den Bürgersteig bzw. den Vorgarten. Das rechte Bild zeigt ein Luftbild des vierten Datensatz und seine Annotationen. Hier sieht man die Gebäude von einem Polygon in Magenta umrandet, das unterste Gebäude ist teilweise von Bäumen verdeckt. Die Handhabung des Annotationswerkzeugs hat uns davon abgehalten, Straßen und Vegetation in

Abbildung 2.13.: Annotationen von zwei Bildern.

Luftbildern zu markieren. Diese werden bei der Bildinterpretation wie Hintergrund betrachtet.

eTRIMS-Benchmark

Wir verwenden als ersten Datensatz, `terra-1` genannt, eine Zusammenstellung von Bildern, die Korč & Förstner (2009) zum Überprüfen von Algorithmen zur Interpretation von Fassadenbildern veröffentlicht haben. Dieser Datensatz wurde bereits bei zahlreichen Versuchen verwendet, wie Veröffentlichungen u. a. von Ripperda & Brenner (2009) und Drauschke & Mayer (2010) belegen.

Der Datensatz zeigt 60 terrestrisch aufgenommene, leicht perspektivisch verzerrte Fassadenbilder aus diversen europäischen Städten, wobei in den meisten Bildern mittelgroße Gebäude aus Basel, Bonn und Heidelberg zu sehen sind. Es wurden alle Instanzen der acht Klassen `building`, `door`, `window`, `car`, `ground`, `pavement`, `sky` und `vegetation` annotiert. Eine Hierarchie der Klassen kommt kaum vor, lediglich die beiden Klassen für Fenster und Tür sind der Gebäudeklasse untergeordnet.

In dem Datensatz wurden insgesamt 142 Gebäude annotiert, wobei jedes der 60 Bilder eine vollständige Fassade zeigt. Die anderen Gebäude sind nur teilweise sichtbar. In den annotierten Fassaden befinden sich 1016 Fenster und 85 Türen. Des Weiteren wurden 67 Autos annotiert sowie 194 Bildbereiche als Vegetation. Die meisten der Vegetations-Annotationen umfassen mehrere Pflanzen, die z. B. in Vorgärten wachsen. 71 der markierten Bildbereiche zeigen Himmel, 76 den Bürgersteig sowie 50 die Fahrbahn vor den fotografierten Gebäuden.

eTRIMS-Originaldaten

Wir verwenden als zweiten Datensatz, `terra-2` genannt, eine Zusammenstellung von 123 Bildern, mit denen wir bereits in (Drauschke, 2009) gearbeitet haben. Auch dieser Datensatz zeigt terrestrisch aufgenommene meist leicht perspektivisch verzerrte Fassadenbilder aus diversen europäischen Städten, wobei sehr viele Bilder Innenstadtfassaden aus Berlin und Hamburg zeigen.

Die originalen Annotationen umfassen 30 Klassen, die im Gegensatz zum vorherigen Datensatz nicht bereinigt wurden. So haben wir einerseits eine große Klassenstruktur zur Modellierung von aggregierten Objekten, andererseits entstanden so auch Klassen mit sehr wenigen Instanzen. Insgesamt wurden 200 Gebäude annotiert, deren 294 Balkone, 227 Türen, 68 Eingänge (die natürlich wiederum die Türen enthalten), 181 sichtbare Dächer, 2609 Fenster (die teilweise zu 220 Fenster-Zeilen gehören), 41 Gauben und 4902 Fensterscheiben als Bestandteile der Fenster. Um die Gebäude herum befinden sich noch 128 Autos sowie 176 Bereiche von Vegetation, 69 Bürgersteige, 37 Fahrbahnen, 118 Bereiche von Himmel und 39 Schilder. Dann gibt es noch weitere Klassen für Balkongeländer, Balkontüren, Treppen, Schornsteine, Tore, Personen etc., auf die wir hier nicht näher eingehen können.

eTRIMS-Auswahl

Im Gegensatz zu den ersten beiden Datensätzen zeigt der dritte Datensatz `terra-3` manuell entzerrte Aufnahmen der Fassaden. Bei der Entzerrung wird die Bildgröße beibehalten und die perspektivisch verzerrte Aufnahme so verkleinert, dass sie vollständig im neuen Bild zu sehen ist. Man erkennt die leichte Perspektive bei den Aufnahmen daran, dass nur ein kleiner Anteil der Pixel des neuen Bildes neu eingefügten Hintergrund zeigt.

Der Datensatz enthält 193 Fassadenbilder aus diversen europäischen Städten, wobei die meisten Bilder Gebäudefassaden aus Basel und Hamburg zeigen. Die Abspeicherung der Homographie ermöglicht es, den eigentlichen Bildinhalt als Viereck zu kennzeichnen. Annotiert wurden alle Objekte von zwölf Klassen. Die Klasse `facade` hat die Unterklassen `balcony`, `door`, `stairs` und `window`, und die Klasse `roof` hat die Unterklassen `chimney`, `dormer` und `window`. Gebäude als Komposition von Fassade und Dach wurden nicht extra annotiert. Als weitere Klassen wurden `ground`, `others`, `sky` und `vegetation` verwendet.

Insgesamt wurden 402 Fassaden mit 354 Balkonen, 557 Türen und 41 Treppen annotiert sowie 392 Dächer mit 57 Schornsteinen und 84 Gauben. Dazu kommen noch 4415 annotierte Fenster, 760 Instanzen der Klasse `others` für Personen, Autos, Straßenlaternen, Mülleimer etc. sowie 272 Bereiche des Bodens, 205 Bereiche des Himmels und 382 Bereiche, die Vegetation zeigen.

Luftbildausschnitte

Den vierten und letzten Datensatz `aerial` haben wir selbst angelegt. Aufbauend auf die von Schuster (2005) und Drauschke *et al.* (2006a) verwendeten Bilder haben wir den Datensatz zusammengestellt und annotieren lassen. Er zeigt 225 Ausschnitte aus Luftbildern des Grazer Vororts Andritz und enthält elf annotierte Klassen.

Im Fokus dieses Datensatzes stehen die Dachstrukturen von Gebäuden. Entsprechend wurden alle 1142 Objekte der Klasse `building` sowie die 984 Objekte der Teilklasse `roof` annotiert. Die 184 besonders prägnanten Dachkomponenten haben wir zudem als Instanzen der Klasse `roof part` gekennzeichnet. Als weitere Teile eines Daches wurden die 1256 Instanzen der Klasse `chimney`, die 169 Instanzen der Klasse `patio` (Dachterasse), die 1587 vielen Dachfenster als `window`, 105 Solarzellen als `solar cell` und 340 Gauben als `dormer` markiert. In der Umgebung von Gebäuden wurden noch die Klassen `car` (1140 mal), `pool` (109 mal) und `shed` (220 mal, für Schuppen und Garagen) annotiert. Auf eine Annotation von Straßen und Vegetation wurde verzichtet, weil man mittels Annotations-Tool von Korč & Schneider (2007) nur Objekte mit einem einfachen Rand markieren kann. Da sowohl Straßen als auch Vegetation sehr viele Bereiche dieser Luftbildausschnitte ausfüllen, wird es folglich häufig zu Regionen der Klasse `background` kommen.

Beispiele

In Abb. 2.14 zeigen wir von jedem Datensatz ein Beispielbild. Links ist die Ansicht eines Reihenhauses aus dem Baseler Vorort Arlesheim zu sehen, die zum Benchmark-Datensatz `terra-1` gehört. Daneben befindet sich die Ansicht eines mittelgroßen Bonner Wohnhauses aus dem Musikerviertel, das im originalen eTRIMS-Datensatz `terra-2` enthalten ist. Als zweites Bild von rechts zeigen wir die entzerrte Aufnahme eines Wohnhauses aus der Innenstadt von München, welches Bestandteil des Datensatzes `terra-3` ist. Die nach der Entzerrung zusätzlich eingefügten Hintergrundpixel haben einen mittleren Grauton. Ganz rechts stellen wir das Luftbild eines Gebäudes aus einem Grazer Vorort dar, die Teil des Luftbild-Datensatzes `aerial` ist. Bei der Visualisierung der Annotationen haben wir deren Reihenfolge manuell festgelegt. Dabei haben wir die Klassenhierarchien berücksichtigt, um zuerst die Annotationen von komplexen Objekten und danach die Annotationen von deren Bestandteilen zu malen.

Wir haben die Bilder nur in einem Fall zufällig aus den Datensätzen ausgewählt. Nur der originale eTRIMS-Datensatz wurde bislang noch nirgendwo veröffentlicht und verwendet. Die anderen Bilder sind schon in einigen Publikationen zur Visualisierung von Ergebnissen verwendet worden. Das linke Bild aus Abb. 2.14 nutzen bereits Drauschke & Mayer (2010) zur Visualisierung ihrer Ergebnisse beim Vergleich verschiedener Texturfilter für die Klassifikation von Pixeln in Gebäudeaufnahmen. Das Münchener Fassadenbild ist als perspektive Aufnahme auch Teil des Benchmark-Datensatzes `terra-1`. Ergebnisse auf diesem Bild zeigen u. a. Ripperda & Brenner (2009) und Jahangiri (2010). Der Luftbildausschnitt wurde von Schuster (2005) erstmals bei einer Publikation verwendet, wir haben ihn aber auch in (Drauschke *et al.*, 2006a) und (Drauschke, 2009) zur Visualisierung unserer Ergebnisse genutzt.

Abbildung 2.14.: Beispielbilder und deren Annotationen aus den vier verwendeten Datensätzen. Nicht markierte Bildbereiche in allen vier Annotationen zeigen wir durch weiße Pixel. Die von der Anzahl der Regionen her sehr wenig vorkommenden Klassen haben wir zu einer neuen Klasse `others` zusammengefasst und visualisieren die entsprechenden Pixel gelb. Bei den weiteren Farben gibt es Unterschiede zwischen den Datensätzen. In der linken Spalte haben wir die Gebäude-Annotationen rot gefärbt, die Vegetation grün und die Fenster blau. In den beiden mittleren Spalten müssen verhältnismäßig viele Klassen visualisiert und unterschieden werden. Die hellgrauen Pixel markieren die Klasse `facade`, die dunkelgrauen Annotationen stellen Fensterscheiben dar und verdecken oft die blau markierten Fenster. Ebenfalls wurden Balkone und Dächer annotiert und in violetten Farbtönen visualisert. In der rechten Spalte zeigen wir einen Luftbildausschnitt, in dem nur drei annotierte Klassen erkennbar sind: Dächer in rot, Autos in grau sowie die neue Klasse `others` in gelb, die vor allem die Klassen zu kleinen Dachstrukturen wie Gauben und Fenster vereint.

3. Hierarchische Segmentierung mit Wasserscheiden

In diesem Kapitel stellen wir unsere Realisierung der hierarchischen Segmentierung eines Bildes vor. Dazu werden wir das Bild im Skalenraum analysieren und mit dem Wasserscheiden-Algorithmus in Regionen einteilen. Durch Definition eines Regionen-Hierarchiegraphen erhalten wir eine hierarchische Anordnung der Regionen und können daraus eine irreguläre Bildpyramide ableiten. Eine Fokussierung auf stabile Regionen in der Pyramide führt zu einer Reduzierung der segmentierten Regionen. In unseren Experimenten analyiseren wir, inwiefern die stabilen Regionen die sich im Bild befindenden Objekte darstellen und ob die Hierarchie der Regionen die Aggregat-Struktur der Objekte wiedergibt.

3.1. Pyramide von Wasserscheidenregionen

Wir beginnen mit einer Beschreibung der hierarchischen Bildsegmentierung. Dazu verknüpfen wir die Bildebenen im diskreten Gauß'schen Skalenraum und die darin segmentierten Wasserscheiden-Regionen zu einer irregulären Bildpyramide. Die verschiedenen Maßstabsebenen des Gauß'schen Skalenraums werden unabhängig voneinander mit dem Wasserscheiden-Algorithmus in Regionen unterteilt. Die Modellierung eines Regionen-Hierarchiegraphen, d. h. ein Graph mit Kanten zwischen Regionen benachbarter Hierarchiebenen, ermöglicht die Verfolgung der Regionen über mehrere Maßstabsebenen hinweg. Auf diese Weise wird unsere irreguläre Pyramide generiert, so dass die detektierten Regionen mit einer präziseren Geometrie beschrieben werden.

3.1.1. Bildsegmentierung im Gauß'schen Skalenraum

Der Gauß'sche Skalenraum ermöglicht die Wahrnehmung von Objekten unterschiedlicher Größe, da in höheren Maßstabsebenen die Details kleinerer Strukturen weggeglättet werden. Wir modellieren den Gauß'schen Skalenraum diskret, d. h. wir glätten das Originalbild mit verschieden großen Gaußfiltern und erzeugen so verschiedene Maßstabsebenen im Maßstabsraum des Bildes. Der Index einer Maßstabsebene wird mit i bezeichnet, der entsprechende Glättungsparameter mit σ_i. Der von uns verwendete Skalenraum besteht aus 41 Maßstabsebenen, wobei diese sehr dichte Diskretisierung des Skalenraums durch die hierarchische Analyse der Regionen begründet wird (siehe Kap. 3.1.2). Die Maßstabsebenen werden logarithmisch zwischen $\sigma_1 = 1$ und $\sigma_{41} = 16$ mit $o = 10$ Ebenen in jeder Oktave angeordnet, d. h.

$$\sigma_i = 2^{(i-1)/o} = 2^{(i-1)/10} \text{ mit } i = 1, ..., 41. \tag{3.1}$$

Die diskreten Maßstabsebenen ermöglichen eine separate Bildsegmentierung mit dem Wasserscheiden-Algorithmus für jede Hierarchieebene. Dazu betrachten wir das Gradientenstärkebild einer Maßstabsebene, da betragsmäßig große Gradientenstärken signifikante Kanten im Bild signalisieren. Weiterführende Angaben zur Vorprozessierung der Bilder und zur Berechnung der Gradientenstärkebildern geben wir in (Drauschke *et al.*, 2006a) an.

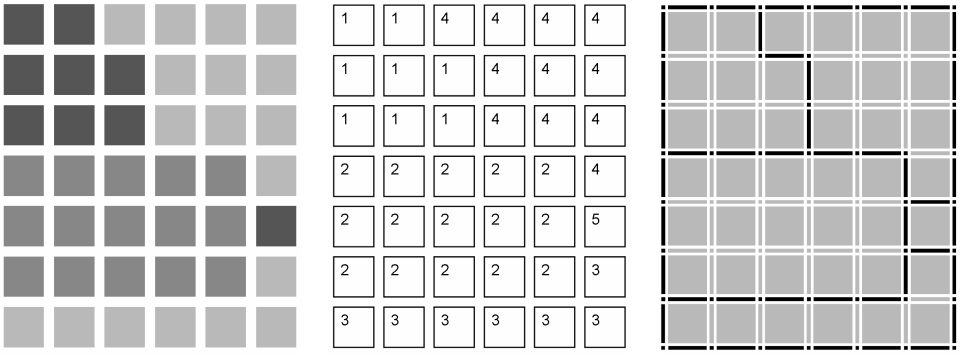


Abbildung 3.1.: Drei Darstellungen einer Bildpartitionierung mit fünf Regionen: grauwertkodiert (links), als Nummernbild (mitte) und mittels Hyper-Raster (rechts).

Die Menge aller segmentierten Regionen wird mit $\mathcal{R}$ bezeichnet. Jede Region erhält eine Identifikationsnummer j, die zusammen mit dem Index i der Maßstabsebene σ_i eindeutig ist. Eine segmentierte Region kennzeichnen wir durch $R_{i,j} \in \mathcal{R}$. Der Wasserscheiden-Algorithmus liefert für jede Maßstabsebene eine vollständige Einteilung des Bildes in sich nicht überlappende Regionen, d. h. in jeder Maßstabsebene wird jedes Bildpixel genau einer Region zugeordnet. Im Gegensatz zur Segmentierung in (Drauschke *et al.*, 2006a) gibt es somit keine Pixel mehr, die den Rand zwischen zwei benachbarten Regionen markieren. Somit entfallen auch problematische Sonderfälle, wie sie Najman & Couprie (2003) zusammengestellt haben.

Zwei Repräsentationsformen haben wir bei der Speicherung der Bildpartitionierung und Regionen umgesetzt. Einerseits speichern wir jede segmentierte Maßstabsebene als Nummernbild, d. h. jedes Pixel enthält die Identifikationsnummer der zugehörigen Region (siehe Abb. 3.1 mitte) statt eines als Intensität interpretierbaren Grauwerts. Zudem gibt es eine zweite Form für die symbolische Bildbeschreibung: das Hyper-Raster, wie es von Winter (1995) vorgeschlagen wurde. In dieser Modellierungsform werden die Ränder zwischen den Regionen durch linienhafte Elemente beschrieben, die zwischen den flächenhaften Bildpixeln verlaufen und sich an den Pixelecken mit anderen Linien schneiden können (siehe Abb. 3.1 rechts). Bei kleinen Regionen, die nur wenige Pixel groß sind, ist die Methode des Nummernbildes sehr effizient, bei großen Regionen mit vielen Pixeln im Inneren der Region ist diese Repräsentationsform mit dem Hyperraster effizienter, insbesondere dann, wenn längere Kanten nur durch Start- und Endpunkt angegeben werden.

3.1.2. Regionen-Hierarchiegraph

Für die Beschreibung von Objekt-Teil-Relationen benötigen wir eine hierarchische Anordnung der segmentierten Regionen, die wir mit einem Regionen-Hierarchiegraph (RHG) modellieren. Diesen bauen wir sukzessive zwischen benachbarten Maßstabsebenen auf, und er enthält ausschließlich gerichtete Kanten, die von einer Region der unteren Maßstabsebene zu einer Region der nächsthöheren Maßstabsebene führen. Der RHG wird als Graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ definiert mit $\mathcal{N} = \mathcal{R}$ und

$$(R_{i,j}, R_{i',j'}) \in \mathcal{E} \Leftrightarrow \left\{ \begin{array}{l} i' = i + 1 \wedge \\ |R_{i,j} \cap R_{i',j'}| > |R_{i,j} \cap R_{i',j''}| \; \forall \, j' \neq j'' \end{array} \right\}, \tag{3.2}$$

wobei die Operation $\cap$ den Bildbereich zurückgibt, in dem sich die beiden Regionen (in eine gemeinsame Bildebene projiziert) überlappen.

Die erste der beiden Bedingungen aus Gl. 3.2 erzwingt die ausschließliche Existenz von Kanten zwischen zwei aufeinanderfolgenden Maßstabsebenen. Die Zuordnung von einer Region zu einer anderen Region der darauffolgenden Maßstabsebene erfolgt über den Flächeninhalt der gemeinsamen Fläche beider Regionen $|R_{i,j} \cap R_{i',j'}|$ als Kriterium. Damit hängt die Entscheidung weder von einem Schwellwert noch vom gewählten Segmentierungsverfahren ab und ist zudem auch bei Subpixel-Verfahren einsatzfähig.

Die Wahl der Diskretisierung der Maßstabsebenen hat Auswirkungen auf den RHG: Die Kantenrelation ist zwar sehr häufig transitiv, aber nicht immer, wie das Beispiel aus Abb. 3.2 zeigt. Dort sind drei aufeinanderfolgende Maßstabsebenen mit vier, drei bzw. zwei segmentierten Regionen zu sehen und der zugehörige RHG, der nach der Gl. 3.2 konstruiert wurde. Transitivität der Relation würde bedeuten, dass $(R_{1,3}, R_{3,2}) \in \mathcal{E}$ gilt, wenn $(R_{1,3}, R_{2,2}) \in \mathcal{E}$ und $(R_{2,2}, R_{3,2}) \in \mathcal{E}$ festgestellt wurde. Ohne die mittlere Maßstabsebene ergäbe sich allerdings $(R_{1,3}, R_{3,1}) \in \mathcal{E}$, da

$$|R_{1,3} \cap R_{3,1}| = 2 > 1 = |R_{1,3} \cap R_{3,2}|.$$

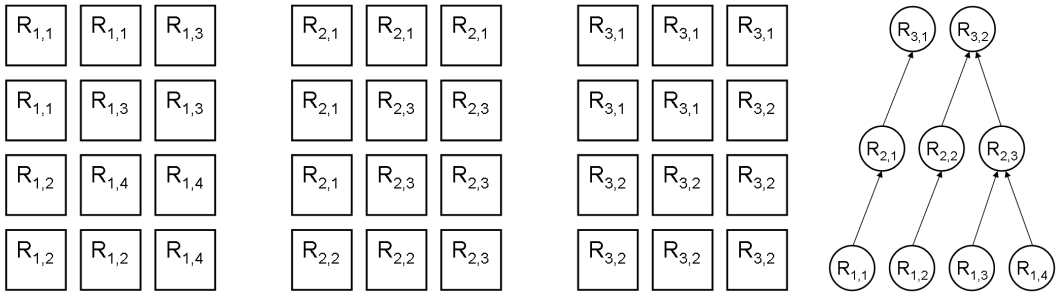


Abbildung 3.2.: Mini-Beispiel für RHG mit drei Maßstabsebenen

Die Kanten des RHG sind gerichtet, d. h. sie führen immer von einer Region aus der Maßstabsebene mit Index i zu genau einer Region der Maßstabsebene mit Index $i+1$. Durch diese Relation können wir somit eine Folge von Regionen definieren und so die Entwicklung der Regionen im Skalenraum mittels RHG beschreiben.

Jede Region (außer die der obersten Maßstabsebene) hat genau eine Nachfolge-Region im RHG. Zur Überprüfung dieser Aussage wird zunächst die Existenz, danach die Eindeutigkeit der Nachfolge-Region gezeigt. Die Ursprungsregion hat eine Fläche, die auf jeden Fall positiv ist. Im von dieser Region abgedeckten Bildbereich befindet sich in der nächsthöheren Maßstabsebene mindestens eine Region, da jede Maßstabsebene vollständig in Regionen zerlegt wird. Unter diesen Regionen suchen wir nun diejenige, deren Schnitt mit dem Bildbereich am größten ist. Damit ist die Existenz ausreichend belegt.

Die Eindeutigkeit wird durch die in Gl. 3.2 enthaltene Ungleichung erzwungen. Allerdings kommt es in der Praxis vor, dass zwei Regionen der höheren Maßstabsebene einen gleichgroßen Schnitt mit der Ursprungs-Region haben. Bei diskreten Segmentierungen kann das bei sehr kleinen Regionen vorkommen, die z. B. nur aus zwei Pixeln bestehen und jedes der beiden Pixel findet sich in der nächsthöheren Ebene in einer anderen Region wieder. In diesen Fällen wird die Region als Nachfolger gewählt, die die niedrige Identifikationsnummer hat. Praktisch gesehen, tritt dieser Fall nur selten auf, da die kleinste Maßstabsebene im Gauß'schen Skalenraum mit $\sigma = 1$ festgelegt wurde. Dort haben die kleinen Regionen bereits eine Größe von 10

oder mehr Pixel. Experimente haben gezeigt, dass die Wahrscheinlichkeit für das Auftreten von zwei gleichberechtigten Nachfolger-Regionen nur 0.0097 ist und damit vernachlässigbar ist.

Durch die Eindeutigkeit der Kanten zu Nachfolger-Regionen nimmt der RHG die Form eines Waldes an, d. h. er besteht aus mehreren Komponenten, und jede Komponente hat eine baumartige Struktur. Bei Umkehrung der Kanten hat jede Komponente des RHG ihre Wurzel in einer Region der höchsten Maßstabsebene.

Lindeberg (1994) hat vier Ereignisse bei der besonderen Entwicklung von Regionen im Gauß'schen Skalenraum benannt: das Zusammenschmelzen von Regionen (*merging*), das Entstehen einer Region (*creation*), das Auflösen einer Region (*annihilation*) und das sich Aufteilen einer Region in mehrere Regionen (*split*).

Der von uns definierte RHG spiegelt nur zwei dieser Entwicklungsmöglichkeiten für Regionen wider. Das Ereignis *annihilation* kann bei uns nicht vorkommen, da für jede Region eine Nachfolger-Region gefunden wird. Ebensowenig kommt das Ereignis *split* vor, da es sonst zwei Nachfolger-Regionen für eine Region geben müsste. Im Gegensatz dazu tritt das Ereignis *merging* in der Struktur des RHG auf, wenn eine Region Nachfolger-Region von zwei oder mehreren Regionen ist. Das Ereignis *creation* kommt vor, wenn eine Region keine Nachfolger-Region einer Region aus der nächstunteren Maßstabsebene ist. Abb. 3.3 zeigt eine Segmentierung im Skalenraum und den daraus extrahierten RHG mit den entsprechenden Ereignissen. Dabei wurden die Regionen einer Maßstabsebene horizontal ausgerichtet, die neu entstandende Region in der zweiten Ebene ist im RHG grau gekennzeichnet.

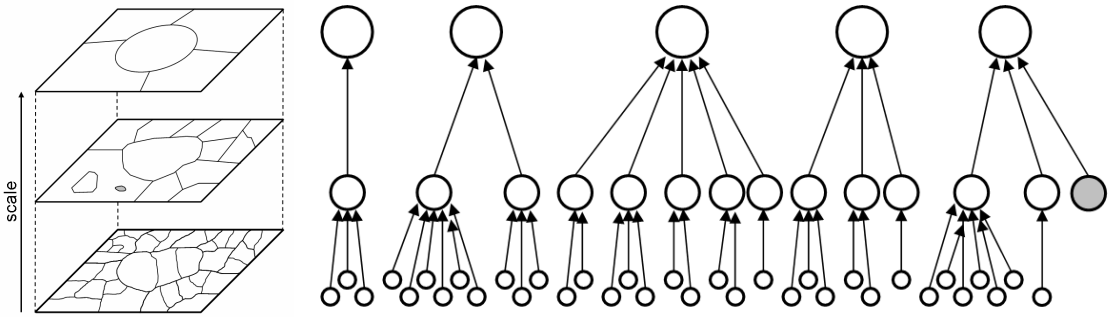


Abbildung 3.3.: Segmentierung im Skalenraum und zugehöriger RHG.

3.1.3. Geometrisch präzise Beschreibung der Regionen

Abb. 3.4 zeigt neben den Originalbildern (obere Reihe) das Bild in der Maßstabsebene mit $\sigma_{31} = 8$ und den Rändern der segmentierten Wasserscheidenregionen in gelb (untere Reihe). Dort ist gut zu erkennen, dass die Glättung mit einem isotropen Gauß-Filter im Skalenraum in einer hohen Maßstabsebene nicht nur zum Verschwinden kleiner Details führt, sondern auch zu unscharfen Kanten und damit ungenau segmentierten Regionen sowie zu abgerundeten Ecken der Regionen. Insbesondere Strukturen wie Fenster nehmen in diesen Segmentierungen (sofern noch durch eine Region detektiert) eher eine ovale als eine rechteckige Form an.

Geometrische präzisere Regionen können im Skalenraum segmentiert werden, wenn man die Entwicklung der Regionen über die Maßstabsebenen hinweg berücksichtigt. Erste Vorschläge über entsprechende Vorgehensweisen im Hinblick auf Kanten beschreiben bereits Witkin (1983) und Bergholm (1987): Wenn die Existenz einer Kante in einer Maßstabsebene

Abbildung 3.4.: Segmentierungen im Gauß'schen Skalenraum.

festgestellt wird, dann kann ihre genaue Lage durch Zurückverfolgen der Kante in die unteren Maßstabsebenen bestimmt werden. Wir wenden diese Methode auf unseren diskreten Skalenraum an und übertragen sie auf segmentierte Regionen.

Bergholm (1987) hat die Kanten über mehrere Maßstabsebenen hinweg durch pixelweisen Vergleich verfolgt. Wir aber betrachten Regionen, die in eine Bildpartitionierung eingebettet sind. Aus diesem Grund muss eine lokale Bewertung zur Verfolgung der Regionen durch eine regionale ersetzt werden. Hierzu verwenden wir den RHG, der die Entwicklung aller Regionen über alle Maßstabsebenen beschreibt. Zusammen mit der Bildpartitionierung der untersten Maßstabsebene ermöglicht er eine Neuaufteilung jeder Maßstabsebene in geometrisch präzisere Regionen $R'_{i,j}$:

$$R'_{i,j} = \cup_k R_{1,k} \text{ es gibt einen Pfad von } R_{1,k} \text{ nach } R_{i,j}, \tag{3.3}$$

wobei ein Pfad eine Folge von aufeinander folgenden Kanten des RHG ist. Da die Kanten im RHG gerichtet sind, setzen wir so die Region $R'_{i,j}$ aus allen Regionen $R_{1,k}$ der ersten Maßstabsebene zusammen, deren Nachfolger die Region $R_{i,j}$ in der i-ten Maßstabsebene ist. Abb. 3.5 zeigt die Segmentierungen der 31. Maßstabsebene im Gauß'schen Skalenraum (obere Reihe) und die neuen geometrisch präziseren Regionen (untere Reihe) im Vergleich. Die Kanten zwischen den Regionen sind nun etwas rauher, dafür sind viele Ecken deutlich rechteckiger.

Durch die Rückverfolgung der Regionen bis zur untersten Maßstabsebene kommt es nicht nur zu einer geometrischen Veränderung der Regionen, sondern auch zu topologischen Veränderungen. Neben möglichen Änderungen in den Nachbarschaftsbeziehungen innerhalb einer Maßstabsebene sind es zwei topologische Änderungen besonders wert, hervorgehoben zu werden. Zum einen können die im Gauß'schen Skalenraum segmentierten Wasserscheidenregionen

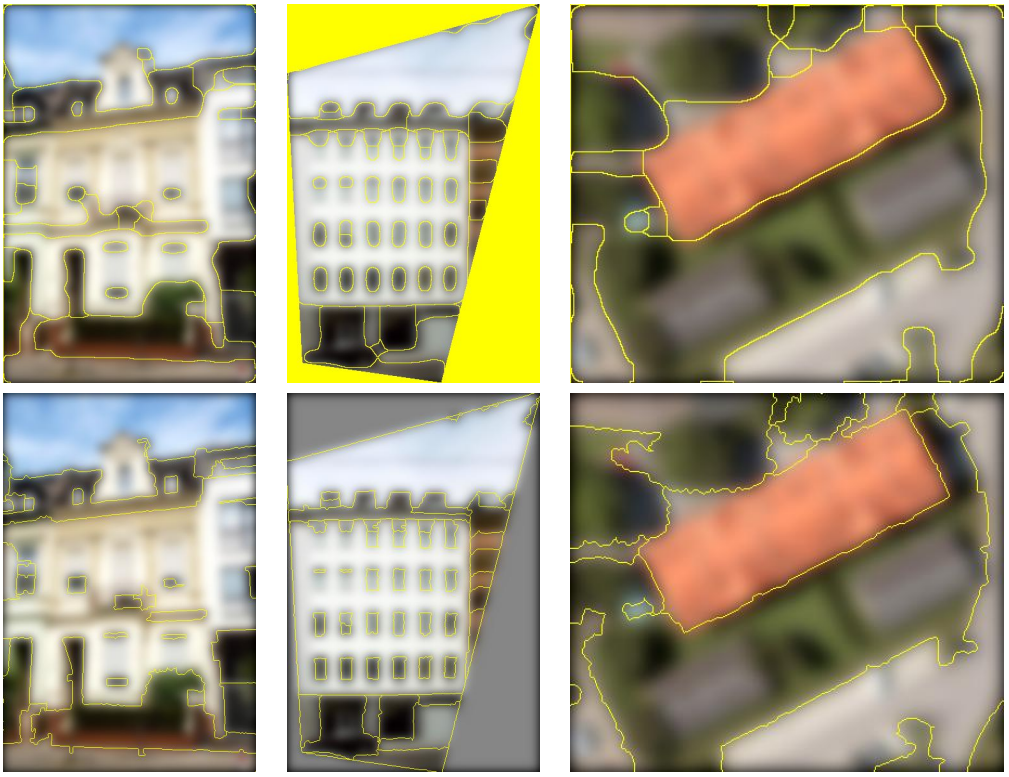


Abbildung 3.5.: Regionen des Gauß'schen Skalenraums in Maßstabsebene mit $\sigma_{31} = 8$ und ihre geometrisch präsizere Beschreibung im Vergleich.

$R_{i,j}$ verschwinden, zum anderen können die bisher aus genau einer Komponente bestehenden Regionen in mehrere Komponenten zerfallen. Eine Region $R_{i,j}$ verschwindet genau dann aus der Segmentierung der i-ten Maßstabsebene, wenn sie nur Nachfolger-Region von Regionen ist, die im Gauß'schen Skalenraum erst in höheren Maßstabsebenen enstanden. Die in Gl. 3.3 formulierte Bedingung gilt dann für keine Region der untersten Maßstabsebene, folglich ist die Vereinigung dieser Regionen zu $R'_{i,j}$ leer. Die Vereinigung der Regionen der ersten Maßstabsebene zu neuen Regionen $R'_{i,j} \in \mathcal{R}'$ garantiert nicht, dass diese eine zusammenhängende Fläche im Bild abdecken. So kann es dazu führen, dass eine Region $R'_{i,j}$ aus mehreren Komponenten entsteht. Dies kann besonders häufig bei fast linienhaften Regionen auftreten, z. B. im Geäst von Bäumen. Experimente haben ergeben, dass topologische Veränderungen mit einer Wahrscheinlichkeit von 0.0336 auftreten.

Da sich Größe und Form der Regionen verändert haben, konstruieren wir erneut einen RHG, wobei wir problemlos die Definition aus Gl. 3.2 wieder verwenden können. Der RHG auf den Regionen $R'_{i,j}$ unterscheidet sich vom RHG auf den Regionen $R_{i,j}$ dadurch, dass alle Knoten und Kanten zu bzw. von diesen Knoten verschwinden, die nicht zugleich Nachfolger von Regionen der ersten Maßstabsebene sind. Der in Abb. 3.3 dargestellte RHG verändert somit nur an einer Stelle die Struktur: Der grau gekennzeichnete Knoten sowie die aus diesem Knoten wegführende Kante kommen im RHG der geometrisch präziseren Regionen nicht mehr vor.

Die geometrische Präzisierung der segmentierten Regionen hat noch einen weiteren für die Darstellung der Segmentierungen wichtigen Vorteil: Bei Verfolgung der Regionen $R'_{i,j}$ im RHG wachsen diese monoton. Auf diese Weise beschreiben die neuen Segmentierungen eine

irreguläre Bildpyramide. In Abb. 3.6 wird diese Struktur gezeigt. Dort wurden die geometrisch präzisen Regionen der Maßstabsebenen mit $\sigma_{11} = 2$ (dünn) und $\sigma_{31} = 8$ (dick) in eine Grafik gezeichnet.

Abbildung 3.6.: Irreguläre Pyramide mit Regionen aus zwei Maßstabsebenen: dick umrandet sind die Regionen aus Maßstabsebene $\sigma_{31} = 8$, dünn die Regionen aus Maßstabsebene $\sigma_{11} = 2$.

3.2. Fokus auf stabile Wasserscheidenregionen

Bislang haben wir nur über die im RHG modellierte hierarchische Struktur der segmentierten Regionen $R'_{i,j}$ erörtert, nicht aber deren Komplexität. Die Segmentierungen im Skalenraum eines 400×600 Pixel großen Bildes mit 41 diskreten Maßstabsebenen erzielen zwischen 1 500

und 3 000 Regionen in der ersten Maßstabsebene sowie 10 und 30 Regionen in der höchsten Maßstabsebene. Unter der Annahme, dass die Anzahl der Regionen über alle Maßstabsebenen mit gleicher Geschwindigkeit abnimmt, enthält die irreguläre Bildpyramide über 30 000 Regionen. Das wird durch Experimente bestätigt: die meisten Bildpyramiden bestehen aus (zum Teil deutlich mehr als) 25 000 Regionen. Eine regionen-basierte Klassifikation mit mehreren hundert Bildern erreicht dabei sehr schnell ihre Grenzen. Aus diesem Grund werden für die weitere Analyse der Regionen besonders markante Regionen, die stabilen Regionen, ausgewählt.

3.2.1. Stabilität als Eigenschaft einer Region

Bei der Charakterisierung der markanten Regionen verwenden wir Wissen über die Erscheinung von Gebäuden in Bildern. Die Objektgrenzen von Gebäuden und Gebäudeteilen sind in Bildern meistens durch deutlich erkennbare Kanten sichtbar. Durch die Prägnanz dieser Kanten können wir sie auch über mehrere Maßstabsebenen hinweg detektieren, und durch den Aufbau einer irregulären Pyramide bleibt außerdem die Lage der Kanten bestehen. Wir erwarten somit, dass sich viele Regionen, die Gebäude und Gebäudeteile zeigen, ihre Lage und Form über viele Maßstabsebenen hinweg nur geringfügig verändern. Dabei wählen wir als Dauer $d = 10$ Maßstabsebenen, d. h. eine vollständige Oktave der Gauß'schen Pyramide.

Die *paarweise Stabilität* zwischen zwei Regionen $R'_{i,j}$ und $R'_{i',j'}$ messen wir mittels der Funktion

$$\Upsilon(i,j,i',j') = \frac{|R'_{i,j} \cap R'_{i',j'}|}{|R'_{i,j} \cup R'_{i',j'}|}, \tag{3.4}$$

die den Wert 1 zurückgibt, wenn die beiden Regionen identisch sind, und 0, wenn die beiden Regionen sich in keinem Pixel überlagern. Wir wenden diese Funktion zur Bewertung der Entwicklung einer Region im Skalenraum an und setzen $i' = i + 1$ bzw. j und j' derart, dass die Region $R'_{i',j'}$ die Nachfolge-Region von $R'_{i,j}$ im RHG ist. Dann beschreibt die Funktion Υ die *lokale Stabilität* der Region $R'_{i,j}$ beim Wechsel zur nächsten Maßstabsebene.

Wenn wir die Stabilität der Region $R'_{i,j}$ nicht lokal, sondern im Vergleich zu einer Folge von Regionen in d anderen Maßstabsebenen bestimmen, dann gibt es $d + 1$ verschiedene Möglichkeiten zur Auswahl dieser Maßstabsebenen. Diese Möglichkeiten erstrecken sich vom Fall, dass alle d Maßstabsebenen unterhalb der i-ten liegen, über die $d - 1$ vielen Fälle, in denen sich die i-te Maßstabsebene in der Mitte befindet, bis zu dem Fall, dass sich alle d Maßstabsebenen oberhalb der i-ten befinden. Innerhalb einer Folge von Regionen markiert der kleinste Wert für eine Überlappung mit der Region $R'_{i,j}$ die instabilste Stelle der Sequenz. Folglich wird die *Stabilität* einer Region $R'_{i,j}$ durch die Funktion Y definiert:

$$Y(i,j) = \max_{k=0\ldots d} \left\{ \min_{i'=i-d+k\ldots i+k} \left\{ \max_{j'} \Upsilon(i,j,i',j') \right\} \right\}. \tag{3.5}$$

Für die Bildinterpretation berücksichtigen wir alle Regionen $R'_{i,j}$, deren Stabilität in d Maßstabsebenen einen Schwellwert übertrifft, d. h. $Y(i,j) \geq \mathfrak{t}$. Setzen wir den Schwellwert $\mathfrak{t} = 1$, dann würden nur diejenigen Regionen der irregulären Pyramide in der weiteren Analyse berücksichtigt werden, die sich innerhalb einer Skalenraum-Oktave um kein einziges Pixel verändern (was praktisch nicht vorkommt). Setzen wir den Schwellwert $\mathfrak{t} = 0$, dann kann jede Region $R'_{i,j}$ das Kriterium erfüllen und es findet keine Auswahl von Regionen statt.

Doch bevor wir unsere Wahl für das Stabilitätskriterium, d. h. den Schwellwerts $\mathfrak{t}$ begründen, wollen wir zunächst erklären, wie wir die Hierarchie der ausgewählten Regionen bestimmen.

3.2.2. Hierarchische Struktur stabiler Regionen

Eine Region mit großer Stabilität über d Maßstabsebenen tritt in der irregulären Pyramide niemals allein auf, sondern ist immer Bestandteil einer Folge von mindestens d Regionen mit großer Stabilität (über d Maßstabsebenen). Da sich diese Regionen nur wenig und teilweise gar nicht unterscheiden, fassen wir diese Folge von Regionen als einen *stabilen Stapel* von Regionen zusammen.

Wenn wir uns den Skalenraum als kontinuierliche 3-dimensionale Struktur vorstellen, dann stellen diese stabilen Stapel prisma-ähnlichen Bereiche dar, deren Höhe parallel zur Maßstabsachse verläuft. Entsprechend nehmen diese Prismen in Querschnitten parallel zur Maßstabsachse eine rechteckige Form an. Die Abb. 3.7 zeigt einen solchen Querschnitt, der auf eine Idee zur Darstellung von Ereignissen im Skalenraum von Witkin (1983) beruht. Jedes dieser weißen Rechtecke repräsentiert eine Folge von Regionen hoher Stabilität, d. h. einen stabilen Stapel. Die horizontale Ausdehnung dieser Rechtecke zeigen die räumliche Ausdehnung der stabilen Stapel im Bild, die vertikale zeigt ihre *Lebensdauer* in der Pyramide an.

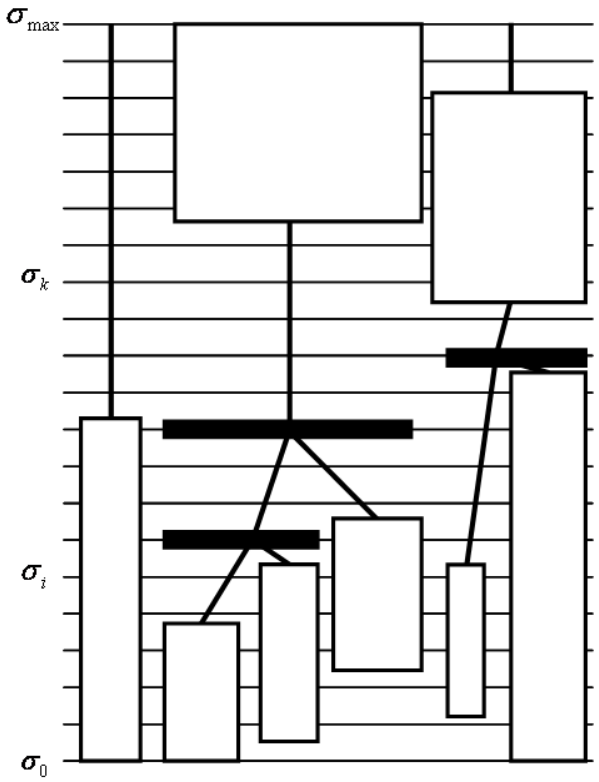


Abbildung 3.7.: Querschnitt durch den Skalenraum parallel zur Skalenachse.

Nach Auswahl der Regionen mit hoher Stabilität müssen wir ihre hierarchische Struktur aus dem RHG der Regionen rekonstruieren, da nur in seltenen Fällen die Region $R'_{i,j}$ die letzte Region eines stabilen Stapels und ihre Nachfolger-Region $R'_{i+1,j'}$ die erste Region eines anderen stabilen Stapels sind. I. d. R. treten zwischen den stabilen Stapeln instabile Zwischenräume auf, die wir bei der Aktualisierung der hierarchischen Struktur überwinden müssen.

Wir betrachten alle Pfade des RHG, die von der letzten Region der stabilen Stapels bis zur höchsten Maßstabsebene führen. Eins der folgenden drei Ereignisse tritt für jeden dieser Pfade

ein: (a) der Pfad trifft in einer Wasserscheidenregion auf mindestens einen anderen Pfad, d. h. die Pfade bestehen ab dieser Region aus denselben Kanten, (b) der Pfad trifft auf Regionen eines anderen stabilen Stapels oder (c) der Pfad führt bis zur obersten Maßstabsebene, ohne dass die anderen beiden Ereignisse eingetreten sind.

Im letzten Fall ist keine hierarchische Struktur vorhanden: Der ursprüngliche Baum des RHG enthält nur einen ausgewählten stabilen Stapel und wird dementsprechend auf einen einzigen Knoten reduziert. Im ersten Fall können die beiden stabilen Stapel hierarchisch angeordnet werden, eventuell ist der Eltern-Stapel Nachfolger mehrerer stabiler Stapel unterer Maßstabsebenen. Beide Ereignisse treten aber verhältnismäßig selten auf, die sukzessive Verschmelzung von stabilen Stapeln des zweiten Falls ist der Regelfall. Im Querschnitt des Skalenraums, siehe Abb. 3.7, werden diese durch Verschmelzung entstandenen Regionen als schwarze Rechtecke visualisert. Sie haben wie die stabilen Stapel in weiß eine räumliche Ausdehnung, hier als Breite dargestellt, haben aber eine Lebensdauer von nur einer Maßstabsebene.

Eine repräsentative Region für jeden stabilen Stapel und jede durch Verschmelzung vormals stabiler Stapel entstandene Region wird von uns zur Bildinterpretation ausgewählt. Sie bilden zusammen die *Menge der stabilen Regionen* $\mathcal{S}$, deren Elemente S_m durch den Index m identifiziert werden. Zusammen mit dieser Menge konstruieren wir den *Wald der stabilen Regionen* (WSR), der aus Bäumen besteht, in denen die stabilen Regionen S_m hierarchisch angeordnet sind. Zwei Knoten des Baums sind durch eine Kante miteinander verbunden, wenn ein Pfad im RHG zwischen Regionen existiert, die durch die Knoten repräsentiert sind. Die Struktur des Walds wird ebenfalls in Abb. 3.7 mit Kanten zwischen den schwarzen und weißen Rechtecken visualisiert, die Pfade zur obersten Maßstabsebene verdeutlichen nur eine Liste, in der wir die Bäume des Waldes sammeln.

Der WSR definiert eine neue hierarchische Struktur, mit der wir die beiden Relationen für Eltern π und Kinder χ (vgl. Kap. 2.1.2) zwischen den stabilen Regionen $S_m \in \mathcal{S}$ beschreiben. Beide Relationen arbeiten auf den Indizes der stabilen Regionen, die leer sind, wenn es keine Eltern-Region (bei den Wurzeln) oder keine Kinder-Regionen (bei den Blättern des Baumes) gibt.

3.2.3. Wahl des Stabilitätskriteriums

Für die Wahl des Stabilitätskriteriums, des Schwellwerts $\mathfrak{t}$, haben wir die Menge der stabilen Regionen $\mathcal{S}$ analysiert. Dazu haben wir auf einem kleinen Datensatz von 50 Fassadenbildern der eTRIMS-Datenbank die Segmentierungen der irregulären Pyramide berechnet und die Anzahl der stabilen Regionen bestimmt, deren Anzahl vom Schwellwert $\mathfrak{t}$ abhängt. Dabei haben wir die zehn verschiedenen Belegungen $0.5, 0.55 \ldots 0.95$ für den Schwellwert $\mathfrak{t}$ berücksichtigt.

Das Experiment hat ergeben, dass die Anzahl der stabilen Regionen $M = |\mathcal{S}|$ sich umgekehrt proportional zu $\mathfrak{t}$ verhält. Unter Verwendung von $\mathfrak{t} = 0.5$ werden im Durchschnitt mehr als 4 000 Regionen ausgewählt, bei $\mathfrak{t} = 0.7$ sind es noch 3 400 Regionen, und bei $\mathfrak{t} = 0.95$ verbleiben nur noch etwas mehr als 1 200 Regionen. Ein großer Teil der stabilen Regionen hat nur eine Fläche von ein paar Pixeln und ist damit deutlich kleiner als das kleinste annotierte Objekt im Bild. Andere Regionen erstrecken sich in nicht annotierten Bildbereichen, zeigen demnach keine für die Auswertung interessanten Objekte. Dem stehen die Regionen gegenüber, die sich partiell oder vollständig mit annotierten Objekten überlagern und daher für die Auswertung interessante Objekte darstellen. Daher koppeln wir die Wahl des Schwellwerts $\mathfrak{t}$ an die semantische Auswertung: Ziel ist es, möglichst viele semantisch relevante Regionen zu erhalten und möglichst wenige, die zu klein sind oder Bildhintergrund beschreiben. Die Funktion $\mathfrak{s}(\mathfrak{t})$ gibt das Verhältnis der semantisch relevanten Regionen zu allen in Abhängigkeit von

$\mathfrak{t}$	$\mathfrak{s}(\mathfrak{t})$
0.5	0.3862
0.55	0.3919
0.6	0.3977
0.65	0.4009
0.7	0.4017
0.75	0.3979
0.8	0.3913
0.85	0.3733
0.9	0.3443
0.95	0.3008

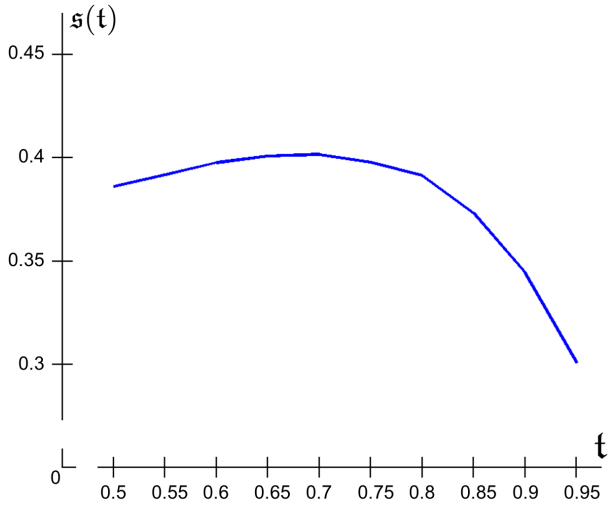


Tabelle 3.1.: Abhängigkeit zwischen dem Schwellwert $\mathfrak{t}$ und dem Verhältnis der semantisch relevanten Regionen zu allen in Abhängigkeit von $\mathfrak{t}$ ausgewählten Regionen $\mathfrak{s}(\mathfrak{t})$.

$\mathfrak{t}$ ausgewählten Regionen an und muss folglich maximiert werden. Die Ergebnisse werden in Tab. 3.1 gezeigt.

In der graphischen Abbildung neben Tab. 3.1 haben wir die Funktion $\mathfrak{s}(\mathfrak{t})$ in Abhängigkeit vom Schwellwert t dargestellt. Es ist deutlich zu erkennen, dass der Graph der Funktion eine konkave Kurve beschreibt, d.h. bei $\mathfrak{t} = 0.7$ hat die Funktion ihr Maximum. Den starken Abfall der Funktion für $\mathfrak{t} > 0.75$ erklären wir uns dadurch, dass dann die Anzahl der stabilen Regionen sinkt: Die Veränderungen der Regionen im Verlauf des Skalenraums, insbesondere der kleinen Regionen, sind zu groß, so dass kein Stapel von Regionen mehr als stabil bewertet wird. Bei $\mathfrak{t} = 0.95$ beträgt das Verhältnis zwischen semantisch relevanten Regionen und allen stabilen Regionen nur noch 30%, vermutlich handelt es dabei um die sehr großen Regionen, die sich nur sehr wenig verändern, um kleine, aber konstraststarke Regionen, die Schattenwürfe oder kleine Objekte zeigen.

Unter der Annahme, dass die Funktion $\mathfrak{s}(\mathfrak{t})$ lokal durch eine Parabel repräsentierbar ist, können wir den maximalen Wert $\mathfrak{t} = 0.684$ bestimmen. Somit berücksichtigen wir die Regionen $R'_{i,j}$ der irregulären Pyramide für die weitere Auswertung, deren Stabilität $Y(i,j)$ mindestens den Wert 0.684 erreicht.

3.3. Experimente

In unseren Experimenten wollen wir unsere ersten beiden Hypothesen überprüfen. Die Erste besagt, dass es möglich ist, eine hierarchische Bildsegmentierung zu erzielen, in der komplexe Objekte und ihre Bestandteile detektiert werden können. Die andere greift die Detektierbarkeit von Objekten und ihren Bestandteilen auf und besagt, dass sich die Relationen der Bestandteilshierarchie des Objekts in prädizierbarer Weise in die Hierarchie der Segmentierung abbilden (vgl. Kap. 1.3).

Abbildung 3.8.: Vergleich zwischen Annotationen und segmentierten Regionen. Oben (v.l.n.r.): Fassadenbild und Annotationen T_l der Klassen Gebäude (ohne Teile), Fenster und Vegetation. Unten (v.l.n.r.): Segmentierte Regionen S_m, deren Target $\widetilde{x}_m$ Mischklasse, Gebäude, Fenster und Vegetation sind.

3.3.1. Validierung der Target-Bestimmung

In Kap. 2.5 haben wir neben dem von uns verwendeten Bildmaterial auch die Annotationen vorgestellt, d. h. die Klassen und ihre Hierarchie aufgelistet. Des Weiteren haben wir in Kap. 2.1.3 unseren Algorithmus vorgestellt, um den segmentierten Regionen das beste Klassenlabel zuzuordnen. Wir haben diesen Ansatz visuell auf einzelnen Bildern überprüft, präsentieren hier aber noch eine quantitative Analyse.

Für jedes Bild erstellen wir klassenspezifische Binärbilder, in denen alle Pixel als Vordergrund dargestellt werden, die zu einer Annotation T_l mit dem entsprechenden Klassenlabel gehören. Analog können wir auch diese klassenspezifischen Binärbilder für die segmentierten stabilen Regionen S_m anfertigen. Im Vergleich zwischen beiden Bildern können wir feststellen, welche der annotierten Pixel auch in segmentierten Regionen zu finden sind, die dasselbe Klassenlabel wie die Annotation erhalten haben.

Fig. 3.8 zeigt neben einem Fassadenbild aus dem Datensatz `terra-1` sieben Binärbilder mit jeweils dem Vordergrund in schwarz. In der oberen Reihe wurden alle Pixel als Vordergrund gezeichnet, die zu einer der Annotationen für `building`, `window` und `vegetation` (v.l.n.r.) gehören, wobei die Pixel, die ebenso zu Objektteilen gehören, ausgeschlossen wurden. Jeweils darunter zeigen die Binärbilder die Pixel der Regionen als Vordergrund, denen unser Algorithmus zur Target-Bestimmung das entsprechende Klassenlabel zugewiesen hat. Ganz links sind die Pixel von Regionen markiert, die zur neu eingeführten Klasse `mixture` gehören, weil sich die Regionen über mehrere Annotationen erstrecken.

Bei dem Bild aus Abb. 3.8 erhalten wir ein Detektionspotential von je 85.0% für Gebäude- und Fensterpixel sowie von 96.4% für Vegetationspixel, wobei *Detektionspotential* den Anteil der Pixel der Annotationen einer Klasse ist, die auch in segmentierten Regionen mit demselben

Klasse	terra-1	terra-2	terra-3	aerial
building	91.7%	76.8%	92.6%	33.9%
car	73.2%	54.4%	-	73.7%
door	72.8%	41.0%	55.0%	-
dormer	-	35.6%	53.4%	63.3%
road / ground	90.8%	72.3%	71.4%	-
roof	-	84.8%	83.6%	81.2%
sky	98.9%	98.8%	97.9%	-
vegetation	90.0%	70.8%	75.4%	-
window	80.5%	63.6%	74.4%	38.1%

Tabelle 3.2.: Klassenspezifische Detektionspotentiale

Klassenlabel liegen. Bei den anderen Klassen erzielen wir ein Detektionspotential von 98.1% für Tür-, 91.7% für Bürgersteig-, 99.3% für Fahrbahn- und 100% für Himmelpixel. Fehler sind mit Verschmelzungen zu erklären, die auf Grund ähnlicher Intensitätswerte basieren, wie z. B. bei der Vegetation in Abb. 3.8, wo der schattige Fassadenbereich unterhalb des Daches mit den angrenzenden Bäumen eine Region bildet. Wegen der größeren Überlappung hat unser Algorithmus dieser Region das Klassenlabel der Vegetation gegeben.

In Tab. 3.2 listen wir das Detektionspotential für wichtige Klassen der vier Datensätze auf. Ein sehr hohes Detektionspotential weisen die Klassen für Gebäude, Vegetation, Himmel sowie Boden bzw. Fahrbahn auf. Schwache Detektionspotentiale entstehen vor allem bei den Objekten, die fast ausschließlich aus Teilen bestehen, wie z. B. `window`-Annotationen im Datensatz `terra-2`, die sehr viele Fensterscheiben haben, oder `building`-Annotationen im Datensatz `aerial`, deren größter Anteil aus Dach besteht.

3.3.2. Detektierbarkeit von Objekten und ihren Teilen

Die vorherige Untersuchung zeigt, dass unser Algorithmus in Kap. 2.1.3 den segmentierten Regionen sinnvolle Targets zuweist. Die Analyse fand aber auf der Pixelebene statt. Dabei wurden alle Pixel eines Klassenlabels ausgewertet. Wenn ein Detektionspotential von 80% festgestellt werden konnte, dann ist dies zwar zufriedenstellend, sagt aber nichts über detektierbare Objekte aus. So könnte z. B. ein Fünftel jeden Fensters fehlen oder jedes fünfte Fenster komplett. Deshalb folgt nun ein Experiment, wo das Detektionspotential auf Objektebene ausgewertet wird.

Ein annotiertes Objekt T_l gilt als *detektierbar*, wenn es mindestens zur Hälfte mit segmentierten Regionen S_m überlappt, die das entsprechende Klassenlabel aufweisen. Die teilweise sehr guten Ergebnisse sind auch dadurch zu erklären, dass wir viele stabile Regionen in den kleinen Maßstabsebenen erhalten. Gerade bei Autos und Gebäuden ist das gesamte Objekt sehr komplex strukturiert: Die vielen kleinen Regionen stellen zusammen ein Großteil des Objekts dar, aber beim Verschmelzungsprozess führt starker Kontrast, z. B. bei Schatten zum Zusammenschmelzen unterschiedlicher Objekte, so dass auf hohen Maßstabsebenen oft das Klassenlabel `mixture` vorkommt.

In Tab. 3.3 stellen wir unsere Ergebnisse bzgl. der Detektierbarkeit von annotierten Objekten vor. Die Detektierbarkeit beruht auf der Überlappung von einer Annotation mit segmentierten Regionen, d. h. es wird nur der Durchschnitt berücksichtigt, wenn die Klassenlabel übereinstimmen. Wir hatten zur besseren Unterscheidung z. B. die neuen Klassen

Klasse	terra-1	terra-2	terra-3	aerial
building	95.0%	93.4%	98.0%	78.3%
car	98.5%	97.6%	-	91.5%
door	89.4%	79.7%	76.8%	-
dormer	-	85.3%	95.2%	65.5%
road / ground	90.0%	83.7%	89.3%	-
roof	-	97.2%	94.3%	85.8%
sky	98.6%	97.4%	97.5%	-
vegetation	94.3%	90.3%	90.0%	-
window	94.3%	91.4%	90.8%	52.5%

Tabelle 3.3.: Detektionspotentiale der annotierten Objekte

facade tiles und facade mixture eingeführt. Bei diesem Experiment werden Regionen mit diesen Klassenlabels zu den facade-Regionen gezählt, weil auch sie Teile der Fassade darstellen (analog building und roof). Da wir sehr hohe Werte für die Detektierbarkeit von annotierten Objekten und ihren annotierten Teilen erhalten, enthält unser Wald der stabilen Regionen brauchbare Regionen für die Interpretation der Szene.

In diesem Experiment haben wir nur die Detektierbarkeit der Objekte und ihrer Teile untersucht. Durch das sukzessive Verschmelzen der Regionen zu größeren gibt es trotz der Fokussierung auf stabile Regionen immer noch sehr viele Regionen, die sich nur geringfügig unterscheiden. Unsere hierarchische Struktur der stabilen Regionen führt demnach in vielen Fällen zu Mehrfachdetektionen.

3.3.3. Visualisierung der Targets

Neben dieser quantitativen Analyse können wir die Target-Bestimmung auch qualitativ untersuchen, indem wir die Targets visualisieren. Diese Aufgabe ist kaum zufriedenstellend zu lösen, da wir Regionen aus 41 verschiedenen Maßstabsebenen berücksichtigen müssen. Wegen der hierarchischen Anordnung überlappen die Regionen sehr stark, wir werden daher die Referenzmaßstabsebene σ_m einer Region S_m als auch ihr Klassenlabel $\widetilde{x}_m$ bei der Reihenfolge der zu markierenden Regionen berücksichtigen. Die Klassen müssen ebenfalls in eine Reihenfolge gebracht werden, um sicherzustellen, dass die komplexen Objekte wie Gebäude vorher im Ausgabebild visualisiert werden als deren Bestandteile, z. B. Fenster. Anderenfalls werden die kleinen Fensteregionen wieder durch Gebäuderegionen verdeckt.

Im Gegensatz zu den vorherigen Untersuchungen, verwenden wir bei der Visualisierung nur noch K statt der urspünglichen $\mathcal{K}$ Klassen für die Targets der Regionen. Wir wollen den Algorithmus für die Visualisierung später bei der Darstellung von Klassifikationsergebnissen wieder verwenden, daher haben wir die nur selten vorkommenden Klassen zu einer neuen Klasse others vereint. Das vereinfacht auch die Darstellung ein wenig, weil wir nun weniger Farben für die Visualisierung der Semantik benötigen.

Den Ablauf dieser Erstellung einer visuellen Bildinterpretation geben wir mit dem Algorithmus 3.1 an. Zuerst initialisieren wir das Ausgabebild als ein weißes Bild. Pixel, die nie zu einer stabilen Region S_m gehören, bleiben weiß und sind damit auch in der symbolischen Bildbeschreibung als solche zu erkennen. Dann wird in Abhängigkeit von der Anzahl der zu visualisierenden Klassen eine Farbcodierung für jede Klasse erzeugt, die wir zuvor manuell bestimmt haben. Dabei haben wir den Kontrast Farben verschiedener Klassen geachtet. So-

weit es möglich ist, stellen wir gleiche Klassen aus verschiedenen Datensätzen einheitlich dar, z. B. Vegetation grün und Fenster blau. Dann wird die Reihenfolge der Klassen bei der Visualisierung bestimmt, da wir große, komplexe Objekte vor deren Bestandteilen darstellen müssen. Anderenfalls werden diese kleinen Objekte von den größeren überlagert und sind somit nicht mehr im Bild sichtbar. Die folgende Reihenfolge haben wir ebenfalls je Datensatz vorher manuell festgelegt. Als erstes werden wir Regionen ins Bild eintragen, die zur Klasse `background` gehören (schwarz), da diese für die Interpretation unwichtig sind. Anschließend markieren wir alle Regionen mit Target `mixture` (sofern noch vorhanden), da diese Regionen mehrere Objekte gleichzeitig zeigen. Anschließend folgen die Gebäuderegionen und alle Regionen von Klassen, die die Umgebung von Gebäude darstellen wie z. B. Vegetation, Himmel, Autos. Danach werden die Regionen im Ausgabebild gekennzeichnet, die zur Klasse `others` gehören. Erst zum Schluss werden die Bestandteile von Gebäuden wie Fenster im Ausgabebild markiert. Die Regionen werden wir nach ihrer Referenzmaßstabsebene σ_m sortiert im Ausgabebild markieren. Wir beginnen bei der höchsten Maßstabsebene, bestimmen alle Regionen dieses Maßstabs, sortieren derart, dass ihre Targets die Reihenfolge der Klassen einhalten, und markieren diese entsprechend im Ausgabebild. Das wiederholen wir für die kleiner werdenden Referenzmaßstabsebene bis wir die niedrigste erreicht haben.

Algorithm 3.1 Algorithmus für Visualisierung der Bildinterpretation.

function VISU-REGIONEN(K, $[T_l]_{l=1}^{L}$, $[S_m]_{m=1}^{M}$, $[\widetilde{x}_m]_{m=1}^{M}$)
  ▷ K: Anzahl der zu visualisierenden Klassen
  ▷ T_l: Annotationen
  ▷ S_m: stabile Region der Bildsegmentierung
  ▷ $\widetilde{x}_m$: passendes Klassenlabel von S_m, d. h. $\widetilde{x}_m \in \{1, \ldots, K\}$
  Erstelle weißes Ausgabebild.
  Erstelle Farb-Codierung für jede der K Klassen
  Bestimme Reihenfolge der Klassen unter Verwendung der T_l
  for $i = \max(\sigma_m), \ldots, \min(\sigma_m)$ **do**
    ▷ Visualisiere Regionen erst aus höheren Maßstabsebenen,
    ▷ danach die aus den kleineren Maßstabsebenen
    for $j = 1, \ldots, K$ **do**
      ▷ ω_j ist Klassenlabel entsprechend der Reihenfolge der Klassen
      Suche alle Regionen mit $\sigma_m = i$ und $\widetilde{x}_m = \omega_j$
      Markiere Pixel dieser Regionen mit passendem Farb-Code im Ausgabebild
    end for
  end for
  return Ausgabebild
end function

Die Visualisierung nach Alg. 3.1 zeigen wir in Abb. 3.9. Mit der Darstellung der Targets sind wir überwiegend zufrieden. Regionen, die Gebäude, Fenster, Vegetation oder Himmel zeigen, haben fast ausnahmslos das richtige Target zugewiesen bekommen. Insbesondere für die beiden Bilder aus Datensatz `terra-1` (linke Spalte) und aus Datensatz `terra-3` (dritte Spalte von links) stellen wir fest: Die Visualisierung der Annotationen und der Targets stimmt meistens überein. Auch bei den Luftbildausschnitt in der rechten Spalten passen die Targets zu den Annotationen. Lediglich die Klasse `others` bedeckt einen viel größeren Bereich im Bild als erforderlich. Für das Bild aus Datensatz `terra-2` ist das ein bisschen anders, da dieser Datensatz sehr viele, sich überlappende Annotationen enthält. In der Mitte des Bildes

befindet sich ein Balkon, der als großes Objekt annotiert wurde, d. h. inklusive des Geländers, der durch ein Rollo verdeckten Tür und eines größeren Bereichs der Fassade drumherum. Das führt zu vielen Komplikationen bei der Suche nach einer passenden Annotation. Daher erhalten die meisten Segmente das Klassenlabel `others`, da dort die Regionen offensichtlich eine Mischung von Klassen zeigen. Die Rollos bei den übrigen Fenstern führen dazu, dass die Segmentierung nur sehr wenige Regionen für Fenster oder Fensterscheiben enthält. Daher haben wir den betreffenden Pixel das Target für Fassade zugewiesen.

3.3.4. Aggregatstruktur von Objekten

Das dritte Experiment an dieser Stelle widmet sich der Frage, ob die hierarchische Struktur der stabilen Regionen auch die annotierte Bestandteil-Beziehung wiedergibt. Dies setzt die Detektion des Objekts als auch dessen Teilen voraus. Wenn wir neben den beiden Regionen (Objekt und Teil) auch einen Pfad in der Hierarchie zwischen beiden Regionen finden, dann werten wir dies als ein gelungenes Abbild der Aggregatsstruktur. Die Gesamtzahl aller Aggregatstrukturen ist die Summe aller Bestandteilsbeziehungen. Die Fassade aus Abb. 3.8 zeigt vier Fenster und eine Tür, also müssen fünf Regionen, die mit den entsprechenden Annotationen überlappen, im Wald der stabilen Regionen gefunden werden, die alle auf einem Pfad zu einer Region liegen müssen, die das gesamte Gebäude beschreibt.

Die Aggregationsraten bei unseren vier Datensätzen haben wir nicht nach Klassen aufgeteilt, da sich das häufig nur auf Gebäude und Dach bezieht. Wir erreichen eine Aggregationsrate von 80.0% beim Datensatz `terra-1` und von 72.8% beim Datensatz `terra-2`. Letzterer hat durch die Annotation von sehr vielen Klassen auch Aggregate wie Balkone, Eingänge, Fenster etc. Die Aggregationsraten beim dritten und vierten Datensatz, `terra-3` und `aerial`, sind 84.6% bzw. 62.8%. Die schlechteren Werte erklären wir uns dadurch, dass die Teile entweder bis zur höchsten Maßstabsebene sichtbar bleiben, ohne mit dem übergeordneten Objekt zu verschmelzen, oder sie verschmelzen mit anderen Objekten, z. B. mit benachbarter Vegetation und bilden dann Regionen der Klasse `mixture`.

3.3.5. Zusammenfassung und Beurteilung

Wir haben in unseren Experimenten dokumentiert, dass viele der manuell erstellten Annotationen durch unsere Segmentierung der stabilen Regionen detektiert werden können. Außerdem bleibt die Aggregatsstruktur, d. h. die Bestandteilshierarchie zwischen den Objekten, in der hierarchischen Struktur der Regionen erhalten. Wir hatten in Kap. 2.2 sechs Kriterien für unsere Bildsegmentierung aufgeführt. Hier haben wir eben den Nachweis dafür gebracht, dass wir verschieden große Objekte segmentieren können und dass deren Anordnung überwiegend die Objektstruktur wiedergibt. Eine präzise Kontur wird durch die Wasserscheidensegmentierung ermöglicht, die Aufnahmesituation spielt keine Rolle. Unsere Segmentierung der stabilen Regionen erfüllt aber nicht die beiden letzten Kriterien. Wir können keine benachbarten Regionen zur Merkmalsgewinnung verwenden, sondern lediglich die Umgebung einer segmentierten Region, und wir haben in unserer hierarchischen Struktur viele Duplikate, was sich aber lediglich auf die Effizienz der Auswertung auswirkt.

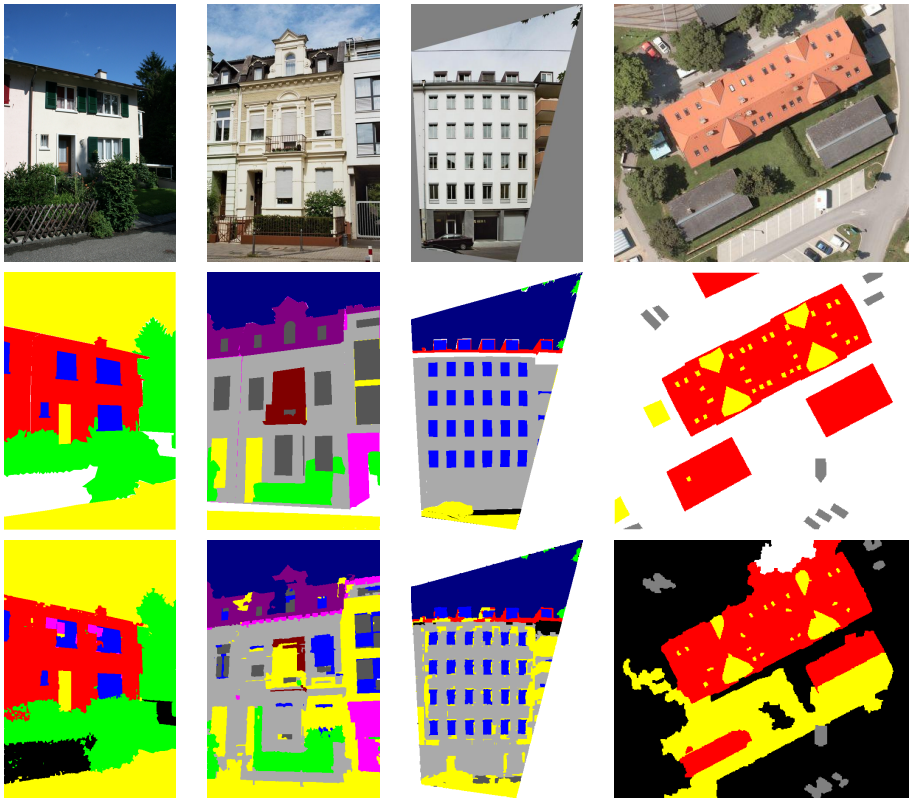


Abbildung 3.9.: Visueller Vergleich zwischen Annotation und Targets der Regionen für Beispielbilder aus Abb. 2.14. In der obersten Zeile zeigen wir wieder die vier Bilder, darunter die Visualisierung der Annotationen dieser Bilder. In der untersten Zeile visualisieren wir die Targets der segmentierten Regionen. Vertikal gesehen bedeuten gleiche Farben auch gleiche Klassen. Weiße Pixel bedeuten dabei, dass dieser Bildbereich nicht annotiert wurde (mittlere Zeile) bzw. dass dort keine stabilen Regionen segmentiert wurden (untere Zeile). Schwarz sind alle Pixel, die zwar als stabile Region segmentiert wurden, zu denen wir aber keine passende Annotation finden können. Pixel von Regionen der zusätzlichen Klasse `others` stellen wir in allen Bildern gelb dar. Bei allen anderen Farben gibt es keine einheitliche Farbgebung bei allen vier Datensätzen. In der linken Spalte zeigen wir das eine Beispielbild aus dem Benchmark-Datensatz `terra-1`. Da haben wir bei den Annotationen und bei den Targets die Gebäudepixel rot eingefärbt, die Vegetation grün und die Fenster blau. Die kleinen magenta-farbenen Regionen stellen Gebäudebereiche dar, die sowohl mit einer Fensterannotation überlappen als auch nicht. Diese Pixel kennzeichnen die Klasse `building-mixture`. In der rechten Spalte zeigen wir einen Luftbildausschnitt, in dem die vier Klassen `background`, `others` (gelb), `roof` (rot) und `car` (grau) erkennbar sind. Da wir hier keine Straßen oder Vegetation annotiert haben, kennzeichnen die Regionen der Klasse `others` vor allem zwei Bereiche: kleinere Dachstrukturen wie Gauben und Fenster sowie Mixturen, die durch die Verschmelzung von Dachregionen mit ihrer Umgebung entstanden sind. In den beiden mittleren Spalten müssen diverse Klassen visualisiert werden. Hier ist vor allem vorzuheben, dass die hellgrauen Pixel zur Klasse `facade` gehören, die dunkelgrauen Annotationen stellen Fensterscheiben dar und verdecken oft die blau markierten Fenster.

4. Klassifikation der stabilen Regionen

In diesem Kapitel thematisieren wir die Klassifikation der stabilen Regionen S_m auf Basis der extrahierten Merkmale $\boldsymbol{F}$. Das Ergebnis der Klassifikation geben wir dabei als K-dimensionaler Vektor aus, dessen Elemente die Wahrscheinlichkeit angeben, mit der die Region S_m zur Klasse ω_k gehört. In unserem Bedingten Bayes-Netz verwenden wir diese Wahrscheinlichkeiten als Zustandsbelegung für die Zufallsvariablen $P(x_m \mid \boldsymbol{F})$.

Für die Klassifikation werden wir für jede stabile Region $D = 65$ Merkmale f_m extrahieren. Die Analyse dieser Merkmale zeigt, dass wir sie durch sehr verschiedene Verteilungen modellieren müssen. Eine Klassifikation mittels Linearer Diskriminanzanalyse (LDA) und Maximum Aposteriori Wahrscheinlichkeit (MAP) wählen wir als Referenz, um die Ergebnisse aus unserem selbst entwickelten Klassifikationsverfahren, die Alternierenden Entscheidungsbäume für K Klassen, vergleichen zu können. Für unsere Evaluation der Klassifikation führen wir eine Kreuzvalidierung durch, bei der wir die Daten so aufteilen, dass wir jede Region aus jedem Bild genau einmal zum Testen verwenden.

4.1. Merkmale von Regionen in terrestrischen Bildern und Luftaufnahmen

Für jede segmentierte Region extrahieren wir nur 65 regionsspezifische Merkmale, damit die Komplexität des Lernalgorithmuses für die Klassifikation überschaubar bleibt. Das d-te Merkmal bezeichnen wir mit f_d. Viele dieser Merkmale wurden bereits in anderen Arbeiten für die Klassifikation von Gebäuden bzw. Gebäudeteilen in Fassaden- oder Luftbildern verwendet. Einige haben wir aber auch selbst ausgesucht. An die Vorstellung der einzelnen Merkmale schließen wir eine Analyse an, inwiefern sich diese Merkmale zur Klassifikation der Regionen eignen.

4.1.1. Bestimmung der Merkmale

Wir haben die Merkmale in fünf verschiedene Gruppen eingeteilt, und zwar in Merkmale zu Form und Größe der Region, Farbmerkmale, Merkmale des Gradientenhistogramms, Merkmale zu einem generalisierten Viereck sowie Texturmerkmale. Wir werden die 65 Merkmale nun kurz vorstellen und gegebenenfalls erörtern.

Zusätzlich zu den Beschreibungen zeigen wir auch diskrete Verteilungen zu zwei ausgewählten Merkmalen. Mit diesen Abbildungen wollen wir die Wirkungskraft des jeweiligen Merkmals demonstrieren, die verschiedenen Klassen ω_k zu trennen. Die Verteilungen haben wir wie folgt berechnet: Zuerst bestimmen wir den minimalen und maximalen Wert eines Merkmals bzgl. des gesamten Datensatz `terra-1`. Dann teilen wir den Bereich zwischen Minimal- und Maximalwert in 100 gleichgroße Intervalle ein und erstellen nach einer zusätzlichen Glättung mit einem Medianfilter klassenspezifische Histogramme, die nach der Normierung die diskreten Verteilungsfunktionen $P(f_d \mid \omega_k)$ darstellen. In diesem Abschnitt zeigen wir die Verteilungen zu den ausgewählten Klassen `building`, `car`, `vegetation` und `window`.

Eine Tabelle mit dem Wertebereich der hier genauer betrachteten Merkmale zeigen wir im Anhang A.1.

Merkmale zu Form und Größe

Die ersten elf extrahierten Merkmale geben Auskunft über Form und Größe der Region S_m. Wir betrachten sie als klassische Merkmale, da sie in vielen Anwendungen und seit vielen Jahren verwendet werden.

Die Anzahl von Komponenten einer Region, die Anzahl der Löcher sowie die Euler-Charakteristik sind unsere ersten Merkmale. In höheren Bildmaßstäben kommt es oft vor, dass z. B. das Geäst eines Baumes als eine Region segmentiert wird, aber bei der Konstruktion der irregulären Pyramide in mehrere nicht benachbarte Komponenten zerfällt. Oder die Fassadenwand wird als eine Region segmentiert, deren Fenster noch als Löcher in der Region existieren.

Die Fläche, der Umfang und der Formfaktor sind ebenso sehr einfach bestimmbare Merkmale. Die verbliebenen fünf Merkmale beschreiben die Bounding Box und umfasst ihre Höhe, ihre Breite, das Verhältnis zwischen der Fläche der Region und der Fläche der Bounding Box sowie die Position des Mittelpunkts der Bounding Box im Bild relativ zu den Abmaßen des Bildes.

Für drei Merkmale zeigen wir deren klassenspezifischen Verteilungen in Abb. 4.1. Links zeigen wir die Verteilung von Merkmal 9, d. h. des Verhältnises zwischen der Fläche der Region und der Fläche der Bounding Box. Das Verhältnis kann Werte zwischen 0 und 1 annehmen, und kann gut durch Normalverteilungen repräsentiert werden. Man kann in der Grafik erkennen, dass diese klassenspezifischen Normalverteilungen leicht varieren. Regionen aus dem Gebäude haben demnach einen höheren Wert als Autos und Vegetation, was vermutlich auf die vielen rechteckigen Strukturen in Fassaden zurückzuführen ist. In der Mitte von Abb. 4.1 zeigen wir die klassenspezifischen Verteilungen des Merkmals 10, der Höhe des Mittelpunkts der Bounding Box im Bild. Wir erkennen gut, dass sich Instanzen der Klasse `building` über die gesamte vertikale Ausrichtung des Bildes verteilen, nur am Bildrand sind es deutlich weniger. Das können wir auf die Aufnahmesituation zurückführen: Bei der Bildaufnahme haben wir meistens die Ansicht so gewählt, dass die Gebäudefassade den größten Teil des Bildes abdeckt. Um dem Gebäude befindet sich im Bild nur noch ein kleiner Rand mit anderen Objekten, z. B. mit Straße und Himmel. Die Instanzen der Klasse `car` befinden sich ausschließlich im unteren Bereich des Bildes, die der Klasse `vegetation` überwiegend im unteren Bereich, was wir auf die Begrünung in den Vorgärten zurückführen. In der vierten Reihe zeigen wir die Verteilung der Klasse `window`, die man gut durch eine Mischverteilung aus drei Gaußverteilungen modellieren könnte. Möglicherweise zeigt sich hier, dass im Datensatz Gebäude mit drei Etagen überwiegen. Rechts in Abb. 4.1 zeigen wir die klassenspezifischen Verteilungen des Merkmals 11, der Breite des Mittelpunkts der Bounding Box im Bild. Hier sehen wir, dass die Gebäude- und Fensterregionen eher in der Mitte vorkommen und Vegetations- und Autoregionen eher am Rand zu finden sind. Auch das führen wir auf die Aufnahmesituation der Bilder zurück.

Farbmerkmale

Die nächsten 14 Merkmale berechnen wir aus den Farbwerten der Pixel jeder Region, wobei die ersten sieben Merkmale die Farbgebung innerhalb der Region und in der Originalauflösung des Bildes charakterisieren, die anderen sieben geben den Betrag der Differenz zur Umgebung an. Zur Bestimmung der Umgebung erweitern wir die Region mittels Dilatation mit einem

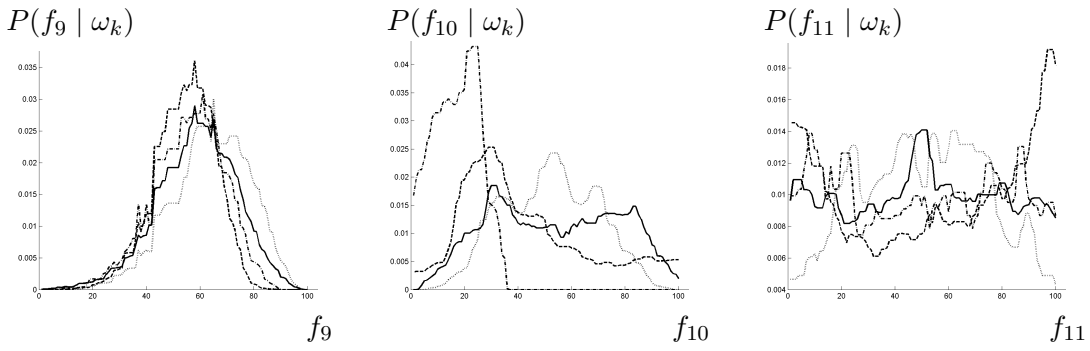


Abbildung 4.1.: In den Abbildungen zeigen wir die Verteilungen von drei Basismerkmalen für die Klassen `building` (durchgezogene Linie), `vegetation` (gestrichelte Linie), `car` (Strich-Punkt-Linie) und `window` (gepunktete Linie). Das Merkmal f_9 stellt das Verhältnis zwischen der Fläche der Region und der Fläche der Bounding Box dar, das Merkmal f_{10} die Höhe und Merkmal f_{11} die Breite des Mittelpunkts der Bounding Box im Bild.

kreisförmigen Strukturelement mit einem sechs Pixel großen Radius und ziehen anschließend die ursprüngliche Region vom Ergebnis wieder ab. Falls Pixel außerhalb des Bildes vorkommen, entfernen wir diese zusätzlich aus der gewonnenen Umgebung.

Unsere Eingangsbilder sind RGB-Bilder, so können wir effizient den Mittelwert und die Streuung eines jeden der drei Kanäle bestimmen. Bei ihren Untersuchungen hat Herms (2007) festgestellt, dass auch der Farbwert (Hue) des HSV-Farbraums ein gutes Merkmal zur Unterscheidung von Fassaden, Fassadenelementen, Vegetation und Himmel darstellt.

Bei der Mittelung des Hue-Wertes ist aber zu beachten, dass dieser zirkulär und nicht (wie die RGB-Werte) linear angeordnet ist. Foley *et al.* (1996) beschreiben die Hue-Werte als Winkel im Intervall $[0, 360)$ [Grad]. Bestimmt man nun das arithmetische Mittel $\mu(\gamma)$ von $\gamma_1 = 10^o$ (rot-orange) und $\gamma_2 = 350^o$ (rot-violett), dann erhält man herkömmlicherweise $\mu(\gamma) = \frac{\gamma_1+\gamma_2}{2} = 180^o$ (türkis), dabei ist 0^o (rot) das gesuchte Mittel $\mu(\gamma)$. Zu beachten bei der Mittelung von Hue-Werten ist, dass das sinnvolle Mittel von zwei Hue-Werten auf dem kürzeren Bogenstück zwischen ihnen liegt.

Wir bestimmen den mittleren Hue-Wert $\mu(\gamma)$ nicht durch Addition der einzelnen Hue-Werte, sondern stellen jeden Hue-Wert γ_i als 2D-Punkt $[\gamma_{i,1}, \gamma_{i,2}]$ auf dem Einheitskreis dar. Dann bestimmen wir den Schwerpunkt aller dieser Punkte mit

$$\begin{aligned} \mu_1(\gamma) &= \mathsf{N}\left(\textstyle\sum_{\mathsf{i}} \gamma_{\mathsf{i},1}\right) \\ \mu_2(\gamma) &= \mathsf{N}\left(\textstyle\sum_{\mathsf{i}} \gamma_{\mathsf{i},2}\right), \end{aligned} \tag{4.1}$$

wobei der Operator N die Normierung durchführt. Aus dem Schwerpunkt $[\mu_1(\gamma), \mu_2(\gamma)]$ berechnen wir abschließend den entsprechenden Winkel auf dem Einheitskreis, den wir als mittleren Hue-Wert interpretieren.

Auch bei der Differenz zwischen dem mittleren Hue-Wert innerhalb der Region und dem mittleren Hue-Wert der Umgebung achten wir darauf, dass das Ergebnis die Distanz auf dem kürzeren Bogenstück angibt. Alle sieben Merkmalsdifferenzen betrachten wir orientiert, so wir einen Wechsel von einer hellen Region zu einer dunklen Umgebung und einen Wechsel von einer dunklen Region zu einer hellen Umgebung durch das Vorzeichen unterscheiden können.

Für drei Farbmerkmale zeigen wir deren klassenspezifischen Verteilungen in Abb. 4.2. Links zeigen wir die Verteilung von Merkmal f_{12}, dem mittleren Rotwert einer Region in der Origi-

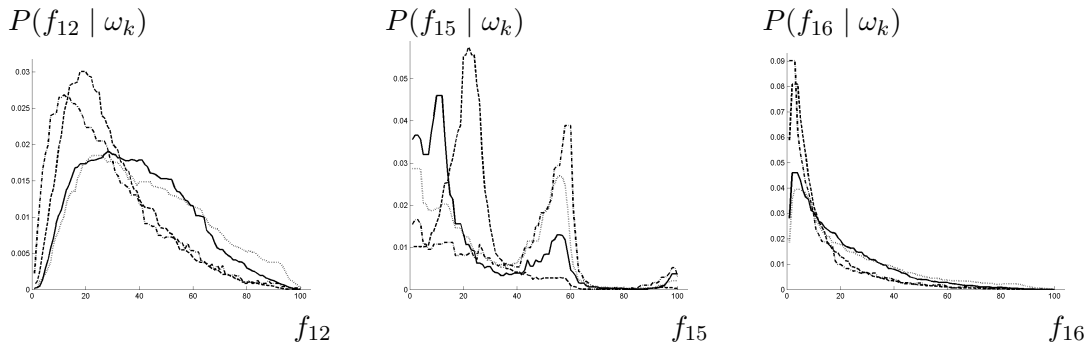


Abbildung 4.2.: In den Abbildungen zeigen wir die Verteilungen von drei Farbmerkmalen für die Klassen `building` (durchgezogene Linie), `vegetation` (gestrichelte Linie), `car` (Strich-Punkt-Linie) und `window` (gepunktete Linie). Das Merkmal f_{12} stellt den mittleren Rot-Wert der Region in der Originalauflösung dar, das Merkmal f_{15} den mittleren Hue-Wert, und Merkmal f_{16} ist die Streuung der Rot-Werte der Region.

nalauflösung des Bildes. Wir erkennen gut, dass Instanzen der Klassen `car` und `vegetation` deutlich kleinere Werte dieses Merkmals haben als Instanzen der Klassen `building` und `window`. Bei der Vegetation liegt es an der hohen Absorption des roten Lichtanteils. Bei den Gebäuden könnten die höheren Rotwerte an hell verputzten Fassaden liegen, bei Fenstern an Spiegelungen und Gardinen. In der Mitte von Abb. 4.2 zeigen wir die klassenspezifischen Verteilungen des Merkmals f_{15}, dem mittleren Hue-Wert der Region in der Originalauflösung des Bildes. Auch hier wird deutlich, dass die Verteilungen sehr verschieden sein können. Gut erkennbar ist die starke Konzentration der Instanzen der Klasse `vegetation` um die grünen Hue-Werte, die ungefähr bei einem Drittel der horizontalen Achse liegen. Den starken Ausschlag für den Hue-Wert 0 (der Balken ganz links in der Abbildung) erklären wir uns an der Umrechnung von RGB-Werten zu einem Hue-Wert: Alle Grautöne zwischen schwarz und weiß werden auf diesen Hue-Wert projiziert, siehe (Foley *et al.*, 1996). Rechts in Abb. 4.2 zeigen wir die klassenspezifischen Verteilungen des Merkmals f_{16}, der Streuung der Rot-Werte der Region. Alle vier Verteilungen könnten wir durch Exponentialverteilungen modellieren, allerdings mit verschieden starkem Abfall der Kurve. Bei Fensterregionen ist die Streuung des Rotwertes größer als bei Regionen aus Autos oder Vegetation. Das liegt vermutlich an den Spiegelungen in den Fensterscheiben.

Merkmale des Gradientenhistogramms

Zwölf Merkmale leiten wir aus den Histogrammen der Gradienten ab. Die Ergebnisse von Korč & Förstner (2008) inspirierten uns zu der Verwendung dieser Merkmalen, wobei sie diese Information zur Klassifikation von Bildbereichen als Himmel (ausschließlich schwache Gradienten), Vegetation (starke Gradienten in mehrere Richtungen) und Gebäude (starke Gradienten in zwei Richtungen) verwenden.

Bevor wir mit der Berechnung der Gradientenhistogramme beginnen, erweitern wir jede Region mittels einer Dilatation mit einem kreisförmigen Strukturelement der Größe 5×5. So können wir auch bei kleinen Regionen aussagekräftige Histogramme herleiten. Dann bestimmen wir für jeden der drei Farbkanäle vier Merkmale der folgenden Art. Es charakterisieren drei Merkmale die Gradienten jeder Region und ein Merkmal gibt Auskunft über den Un-

terschied zwischen der Region und ihrer Umgebung (analog zu den Farbmerkmalen). Für die Region und ihre Umgebung berechnen wir jeweils ein gewichtetes Histogramm, wobei die Gradientenstärke an einem Pixel das Gewicht des Histogrammeintrags liefert. Somit haben große Gradienten viel Einfluß auf das Histogramm.

Die Orientierung des Gradienten an einem Pixel liefert die Position im Histogramm, wobei jede Position ein Intervall von 45 Grad abdeckt. Die mittleren Winkel dieser Intervalle zeigen in die acht Hauptrichtungen eines Kompasses. Wir erwarten, dass sich die Histogramme besonders gut für die Klassifikation von Regionen aus entzerrten Bildern eignen, da hier die stärksten Kanten in horizontaler und vertikaler Richtung verlaufen. Dadurch erwarten wir zwei starke Modalwerte der Histogramme. Bei den Luftbildern sind Strukturen mit deutlichen Kanten wie Straßen und Gebäude in allen Richtungen vorhanden, dadurch eignen sich diese Histogramme vermutlich wenig für die Klassifikation in Luftbildern. Eine Normalisierung der Orientierungen der Gradienten nach der stärksten Richtung im Bild haben wir nicht verfolgt, weil durch die Struktur der Gebäude zwei Richtungen nahezu gleichwertig und daher kaum automatisch sinnvoll zu favorisieren sind.

Die Histogramme normalisieren wir und dann bestimmen wir die drei Positionen mit den höchsten drei Einträgen. Die Differenz zwischen dem höchsten und dem zweithöchsten Wert sowie die Differenz zwischen dem höchsten und dem dritthöchsten Wert werden als Merkmale verwendet, ebenso die Entropie des normalisierten Histogramms. Der Unterschied zur Entropie des normalisierten Histogramms der Umgebung ist das vierte Merkmal, das wir zu jedem Farbkanal bestimmen.

Die Verteilungen der Merkmale des roten Kanals zeigen wir in Abb. 4.3. Von links nach rechts zeigen wir dabei die Verteilungen für die Differenz zwischen dem maximalen Histogrammwert und dem zweitgrößten, die Differenz zwischen dem maximalen Histogrammwert und dem drittgrößten, die Entropie des Gradientenhistogramms sowie den Unterschied zwischen der Entropie des Gradientenhistgramms der Region zur Entropie des Gradientenhistogramms der Umgebung der Region. In den beiden linken Abbildungen erkennen wir gut, dass die Differenzen zwischen dem maximalen und zweit- bzw. dritthöchsten Histogrammwert bei Vegetation besonders klein sind. Wie schon Korč & Förstner (2008) bemerkten, sind die Gradienten in Vegetationsbereichen sehr unregelmäßig. Bei Gebäuden und Fenstern überwiegen zwei Hauptrichtungen, weshalb die Differenzen zum dritthöchsten Histogrammwert bei diesen beiden Klassen größer ausfallen als bei der Vegetation. Entsprechend können wir auch die Verteilungen bei Merkmal f_{28} erklären: Die Entropie ist hoch bei der Vegetation, wo die Gradientenhistogramme wegen der unregelmäßigen Struktur eher gleichverteilt sind.

Merkmale eines generalisierten Vierecks

Weitere zwölf Merkmale bestimmen wir aus einer Generalisierung der Region S_m. Wir approximieren den Rand der Region durch ein Viereck. Dieses konstruieren wir wie folgt: Von einem beliebigen Randpunkt P_0 der Region S_m bestimmen wir zuerst den Punkt P_1, der den größten Abstand zu P_0 hat. Danach bestimmen wir den Randpunkt P_3, der den größten Abstand zu P_1 hat. Dadurch ist die Strecke $\overline{P_1P_3}$ eine der längsten Strecken zwischen zwei Randpunkten der Region. Sie zerteilt die Region S_m in zwei Teile, und in jedem dieser beiden Teile suchen wir die Randpunkte P_2 bzw. P_4, die sich am weitesten von der Strecke $\overline{P_1P_3}$ entfernt befinden. Die vier Punkte P_1, P_2, P_3 und P_4 bilden ein Viereck, das eine gute Approximation von S_m darstellt.

Abb. 4.4 zeigt die Kontur einer Region (in schwarz) sowie den beliebig gewählten Startpunkt P_0 für die Viereckbestimmung. Der Eckpunkt von S_m mit dem größten Abstand zu P_0 ist P_1,

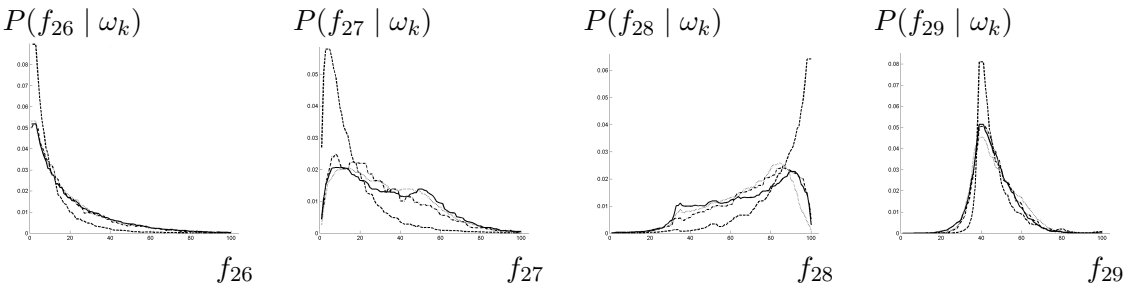


Abbildung 4.3.: In den Abbildungen zeigen wir die Verteilungen der Merkmale des normalisierten Gradientenhistogramms des roten Kanals für die Klassen `building` (durchgezogene Linie), `vegetation` (gestrichelte Linie), `car` (Strich-Punkt-Linie) und `window` (gepunktete Linie). Das Merkmal f_{26} stellt die Differenz zwischen dem maximalen Histogrammwert und dem zweitgrößten dar, Merkmal f_{27} die Differenz zwischen dem maximalen Histogrammwert und dem drittgrößten. Des Weiteren zeigen wir mit Merkmal f_{28} die Verteilung der Entropie der Gradientenhistogramme und mit Merkmal f_{29} den Unterschied zwischen der Entropie des Gradientenhistgramm der Region zur Entropie des Gradientenhistogramms der Umgebung der Region.

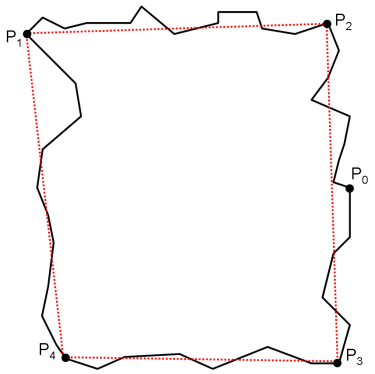


Abbildung 4.4.: Generalisiertes Viereck (rot) der Region S_m und beliebig gewählter Startpunkt P_0 für die Viereckbestimmung.

und P_3 hat den größten Abstand zu P_1. Mit Hilfe des Algorithmus von Douglas & Peucker (1973) bestimmen wir die beiden zusätzlichen Eckpunkte P_2 und P_4 und erhalten so eine Generalisierung von S_m in Form eines Vierecks (in rot).

Aus diesem generalisierten Viereck werden nun die Merkmale abgeleitet. Die drei Elemente der symmetrischen Momentenmatrix der Figur liefern die ersten drei Merkmale. Die beiden Eigenwerte dieser Momentenmatrix, ihr Verhältnis (kleinerer zu größerer) und die Orientierung der Hauptachse die nächsten vier Merkmale. Das Verhältnis zwischen der Anzahl der Pixel im Schnitt der Region und dem Viereck und der Anzahl der Pixel, die der Region oder dem Viereck gehören, verwenden wir als weiteres Merkmal. Als die letzten vier Merkmale dieser Art bestimmen wir die Winkel der vier Ecken des Vierecks. Diese ordnen wir so an, dass wir mit dem größten Winkel beginnen, dann folgen die anderen Winkel im Uhrzeigersinn. Wir erwarten, dass diese Merkmale besonders gut geeignet sind, um insbesondere viereckige Objekte wie Fenster, Türen und Fassaden von anderen zu unterscheiden.

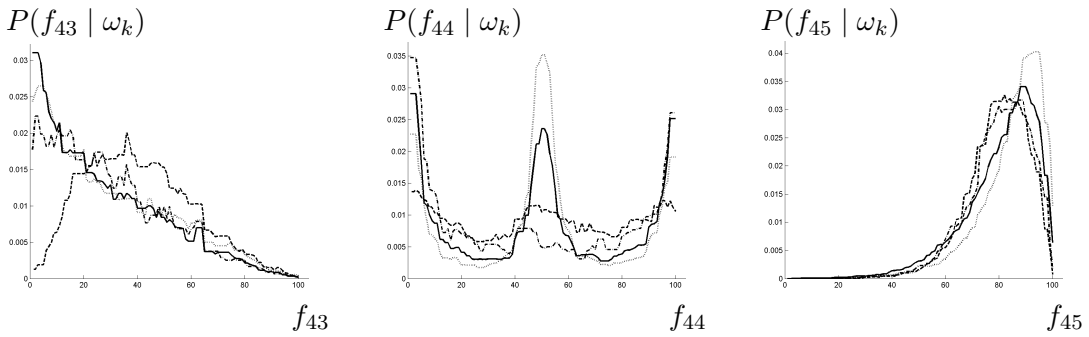


Abbildung 4.5.: In den Abbildungen zeigen wir die Verteilungen von drei Merkmalen des generalisierten Vierecks für die Klassen `building` (durchgezogene Linie), `vegetation` (gestrichelte Linie), `car` (Strich-Punkt-Linie) und `window` (gepunktete Linie). Merkmal f_{43} beschreibt das Achsenverhältnis des Vierecks, Merkmal f_{44} die Orientierung der Hauptachse des Vierecks sowie Merkmal f_{45} das Verhältnis zwischen der Anzahl der Pixel im Schnitt der Region und dem Viereck und der Anzahl der Pixel, die der Region oder dem Viereck gehören.

Die Verteilungen von drei der eben aufgezählten Merkmale zeigen wir in Abb. 4.5. Links zeigen wir die klassenspezifischen Verteilungen zu Merkmal f_{43}, dem Achsenverhältnis des Vierecks. Fenster- und Gebäuderegionen haben häufiger kleine Werte als Vegetationsregionen. Das könnte daran liegen, dass viele Vegetaionsregionen kompakt sind, viele Fenster- und Gebäuderegionen dagegen eher eine längliche Form haben. In der Mitte von Abb. 4.5 zeigen wir die Verteilungen von Merkmal f_{44}, der Orientierung der Hauptachse des Vierecks. Die Spitzen am Rand und bei 90 Grad im Zentrum der Verteilungen zeigen, dass viele Regionen eine horizontale oder vertikale Ausrichtung haben. Diese Erscheinung ist bei Autoregionen am wenigsten ausgeprägt, bei Fensterregionen am stärksten. Rechts in der Abbildung zeigen wir die Verteilungen von Merkmal f_{45}, das das Verhältnis zwischen der Anzahl der Pixel im Schnitt der Region und dem Viereck und der Anzahl der Pixel, die der Region oder dem Viereck gehören, anzeigt. Je näher der Wert an 1 liegt, desto mehr ähnelt die Region dem generalisierendem Viereck. Hier erkennen wir gut, dass Fensterregionen öfter hohe Werte erreichen als andere Regionen.

Texturmerkmale

Die restlichen 16 von uns extrahierten Merkmale einer Region S_m bezeichnen wir als Texturmerkmale. Drauschke & Mayer (2010) haben verschiedene Texturfilter im Hinblick auf ihre Wirksamkeit bei der pixelbasierten Klassifikation in Fassadenbildern untersucht. Laws (1980) hat eine diesbzgl. sehr erfolgreiche Menge von Filtern entwickelt, aber auch die Filter, die auf den Arbeiten von Haar und Walsh basieren (Petrou & Bosdogianni, 1999), waren ähnlich gut.

Wir verwenden die Haar-Filter, da sie die kleinste Menge dieser drei besonders erfolgreichen Filtermengen darstellt. Sie werden aus den orthogonalen Funktionen bestimmt, die Haar (1910) als gute Menge zur Approximation von beliebigen Funktionen vorgeschlagen hatte. Wir haben die ersten vier Funktionen durch vier Stützstellen diskretisiert (siehe Abb. 4.6), die Variable ж gibt dabei den Definitionsbereich für das Bildsignal an. Aus den eindimensionalen Funktionen können wir 16 zweidimensionale Filter konstruieren, siehe Abb. 4.7 links. Aus Komplexitätsgründen haben wir unsere Haar-Filter auf maximal zweite Ordnung beschränkt.

Die Basisbilder der Haar-Filter bis zur zweiten Ordnung haben wir wie üblich als Binärbilder

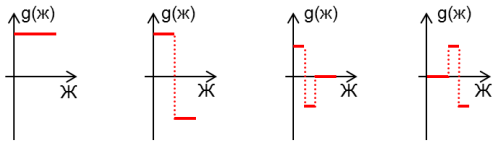


Abbildung 4.6.: Eindimensionale Haar-Funktionen.

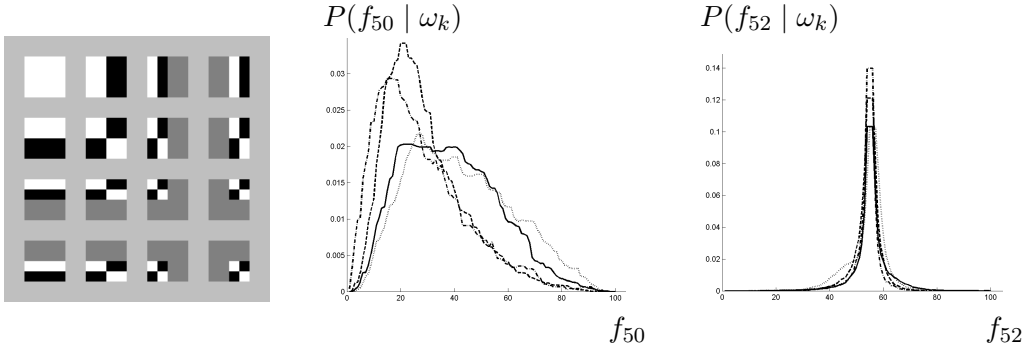


Abbildung 4.7.: Links: Basisbilder der Haarfilter. Die Filter werden zeilenweise den Merkmalen 50 bis 65 zugeordnet. In der Mitte und rechts zeigen wir die Verteilungen von den beiden Texturmerkmalen 50 und 52 die zwei Merkmalen spezifische Verteilungen für die Klassen `building` (durchgezogene Linie), `vegetation` (gestrichelte Linie), `car` (Strich-Punkt-Linie) und `window` (gepunktete Linie). Merkmal f_{50} gehört zum ersten Filter und beschreibt den mittleren Grauwert der Region, das Merkmal f_{52} gehört zum dritten Filter in der ersten Reihe der Basisbilder.

visualisiert. Allerdings stehen die weißen Filterelemente für positive Einträge, die schwarzen für negative. Das mittlere Grau zeigt 0-Einträge an. Diese Art von Filter haben Viola & Jones (2001a) in ihren sehr effizienten und sehr präzisen Gesichtsdetektor integriert, so dass diese Filter auch in vielen anderen Anwendungsbereichen eingesetzt werden. Analog zu den Abbildungen in den vorangegangenen Abschnitten zeigen wir auch für zwei Texturmerkmale die klassenspezifischen Verteilungen, siehe Abb. 4.7 Mitte und rechts. Die Verteilungen der anderen differenzbasierten Texturfilter ähneln den Verteilungen des Merkmals 52.

4.1.2. Charakterisierung der Merkmale

Im Anschluss an die Merkmalsbestimmung analysieren wir die Merkmale auf dem Datensatz `terra-1`. Dabei gehen wir zuerst auf qualitative Aussagen ein, anschließend untersuchen wir die Unterscheidbarkeit der Klassen auf Grund dieser Merkmale. Dazu verwenden wir die Hellinger-Distanz zwischen den klassenspezifischen Verteilungen.

Qualitative Analyse

Von den 65 Merkmalen sind drei diskret, nämlich die Anzahl der Komponenten der Region, die Anzahl der Löcher und die aus beiden bestimmte Euler-Zahl. Neben den ersten beiden Merkmalen gibt es noch 35 weitere Merkmale, die ausschließlich nicht negative Werte haben. In einem weiteren Schritt haben wir Histogramme der einzelnen Merkmale bestimmt und diese

zu diskreten Verteilungen umgeformt, um einen Eindruck zu erhalten, welche Verteilungen diese Merkmale haben.

Typische Verteilungen mit 100 Werten zwischen den kleinsten und den größten Merkmalsausprägungen haben wir in den beiden Abb. 4.8 und 4.9 dargestellt. Zur Bestimmung der möglichen Wahrscheinlichkeitsverteilung haben wir sie mit den Beispielfunktionen aus (Papoulis & Pillai, 2002) verglichen. Alle anderen Merkmale, deren Wahrscheinlichkeitsverteilungen wir hier nicht dargestellt haben, zeigen große Ähnlichkeiten zu den hier aufgeführten Fällen.

Abb. 4.8 (a) zeigt die Verteilung der Anzahl der Komponenten, dessen größter Wert 114 ist. Über 95% der Regionen bestehen aus nur einer Komponente, so dass das Histogramm aus lediglich einem Balken zu bestehen scheint. Dieses Merkmal würden wir durch eine Poisson-Verteilung modellieren. Abb. 4.8 (b) stellt die Höhe der Bounding Box einer Region dar, wir würden es durch eine Exponentialverteilung charakterisieren. Das Merkmal bei (c) stellt das Verhältnis der Flächen zwischen Region und Bounding Box dar, das wir durch eine Gauß-Verteilung modellieren könnten. Die beiden Merkmale in Abb. 4.8 (d) und (e) geben die Position des Schwerpunkts der Bounding Box im Verhältnis zu den Seitenlängen des Bildes wieder. Teil (d) zeigt die Höhenposition, (e) die Breitenposition. Während die Breite nahezu gleichverteilt ist, kann man die Verteilung des Höhenmerkmals gut mit der speziellen Aufnahmekonfiguration von Fassadenbildern begründen. Die Bilder zeigen in unteren Bildbereichen meistens geteerte Straßen und am oberen Bildrand Himmel. Dort werden durch die Homogenität der Objekte weniger Regionen segmentiert als in der Bildmitte, wo sich heterogene Fassaden und umrandende Vegetation befinden. Wir schlagen vor, die Höhenposition durch eine χ^2-Verteilung zu repräsentieren. Das Merkmal aus Abb. 4.8 (f) zeigt eine der Ausgaben durch die Texturfilter, von denen 15 sehr ähnlich aussehen und durch Gauß-Verteilungen modelliert werden können.

In Abb. 4.9 zeigen wir die Wahrscheinlichkeitsverteilungen von zwölf weiteren Merkmalen. Für die Modellierung der Verteilungen dieser Merkmale schlagen wir folgende vor: β-Verteilungen für (g), (i) und (k), eine Mischung aus β- und Gauß-Verteilung für (j), χ^2-Verteilung für (a), eine Mischung aus χ^2- und Gauß-Verteilung für (f), eine Exponentialverteilung für (c) und (e), eine Mischung aus Expontial- und Gauß-Verteilung für (h) sowie eine Gauß-Verteilung für die beiden Merkmale (d) und (l). Eine Verteilung für Merkmal aus (b) würden wir durch eine Mischverteilung aus mehreren Gauß-Verteilungen bestimmen.

Quantitative Analyse

Zur quantitativen Analyse bzgl. der Unterscheidbarkeit der Klassen auf Grund der von uns extrahierten Merkmale bestimmen wir die Hellinger-Distanzen zwischen den klassenspezifischen Verteilungen. Eine Formulierung der Hellinger-Distanz $\mathcal{H}_d$ bzgl. des Merkmals f_d für zwei diskrete Verteilungen $P(f_d \mid \omega_k)$ und $P(f_d \mid \omega_{k'})$ entnehmen wir (Ngom & Emilion, 2008):

$$\mathcal{H}_d(P(f_d \mid \omega_k), P(f_d \mid \omega_{k'})) = \frac{1}{2} \sum \left(\sqrt{P(f_d \mid \omega_k)} - \sqrt{P(f_d \mid \omega_{k'})} \right)^2, \tag{4.2}$$

wobei die Summe über die von uns berechneten 100 Intervalle der Verteilungen läuft. Wir haben die Hellinger-Distanz durch den zusätzlichen Faktor normiert, so dass die größtmögliche Distanz $\mathcal{H}_d = 1$ ist, wenn die Eingabegrößen Wahrscheinlichkeitsverteilungen sind. Diese größtmögliche Distanz $\mathcal{H}_d = 1$ bedeutet: Die beiden Verteilungen sind perfekt trennbar, d. h. ein Maximum-Likelihood-Klassifikator könnte diese beiden Klassen bzgl. dieses Merkmals fehlerfrei unterscheiden.

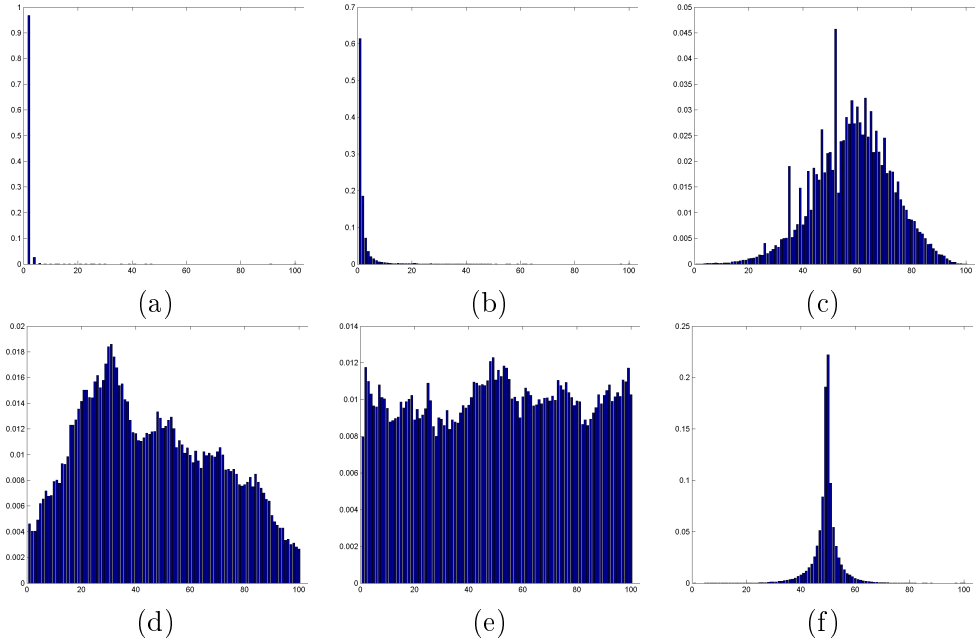


Abbildung 4.8.: Wahrscheinlichkeitsverteilungen ausgewählter Merkmale: (a) für die Anzahl der Komponenten einer Region, (b) für die Höhe der Bounding Box einer Region, (c) für das Verhältnis der Flächen von Region und Bounding Box, (d) für die Höhenposition des Schwerpunkts der Bounding Box bzw. (e) dessen Breitenposition sowie (f) für einen Texturfilter.

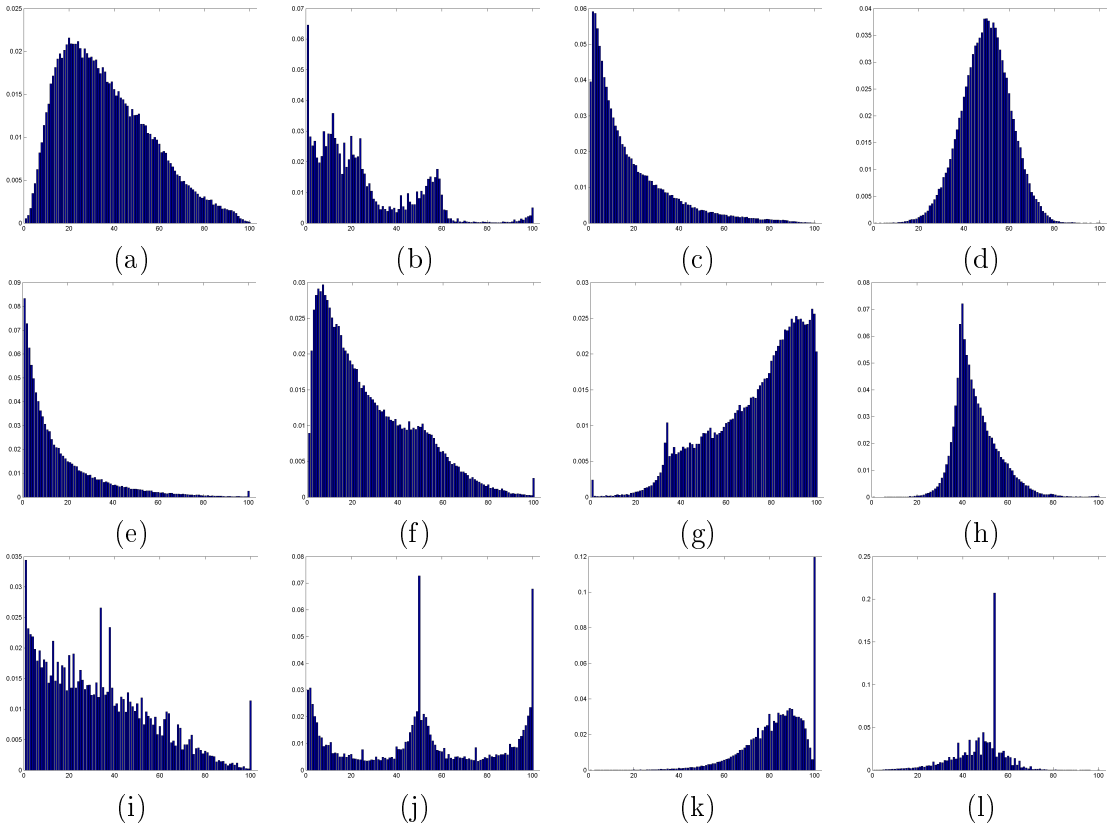


Abbildung 4.9.: Wahrscheinlichkeitsverteilungen einiger Merkmale, i. d. R. des Rot-Kanals: (a) für den Farb-Mittelwert, (b) für den Mittelwert des Hue-Werts, (c) für die Farb-Streuung, (d) für den Mittelwert der Differenzen zwischen Region und Umgebung, (e) für das Verhältnis zwischem Maximalwert und zweithöchsten Wert des normalisierten Gradienthistogramm, (f) analog dazu für das Verhältnis zwischem Maximalwert und dritthöchsten Wert, (g) die Entropie des normalisierten Histogramms sowie (h) für die Differenz der Entropien der Histogramme zwischen Region und Umgebung, (i) dem Achsenverhältnis des Vierecks, (j) der Orientierung des Vierecks, (k) das Verhältnis zwischen Pixeln der Region und denen des Vierecks sowie (l) des zweiten Winkel.

Tabelle 4.1.: Hellinger-Distanzmatrix zwischen sieben Klassen

	building	car	vegetation	window	build. mixt.	background
others	0.238	0.183	0.181	0.403	0.382	0.154
building		0.532	0.158	0.056	0.045	0.381
car			0.357	0.664	0.675	0.135
vegetation				0.177	0.182	0.254
window					0.041	0.505
build. mixt.						0.504

Wir sind an der besten Unterscheidung zwischen zwei Klassen interessiert, daher definieren wir die Hellinger-Distanzmatrix $\mathcal{H}$ unter Berücksichtigung zweier Klassen ω_k und $\omega_{k'}$ durch die größtmögliche merkmalsspezifische Hellinger-Distanz:

$$\mathcal{H}(k, k') = \max_d \mathcal{H}_d(P(f_d \mid \omega_k), P(f_d \mid \omega_{k'})). \tag{4.3}$$

Für $k = k'$ ergibt sich wegen der Definition der Distanz $\mathcal{H}(k, k) = 0$, die anderen Distanzen $\mathcal{H}(k, k')$ stellen wir in Tab. 4.1 zusammen. Besonders hohe Hellinger-Distanzwerte können wir zwischen den Verteilungen der Klassen Gebäude und Auto, zwischen Auto und Fenster, zwischen Auto und Gebäudeteil-Mixturen sowie zwischen Fenster und Hintergrund und zwischen Gebäudeteil-Mixturen und Hintergrund feststellen, denn in diesen Fällen hat die Hellinger-Distanzmatrix Werte von 0.5 oder größer. Schwierig könnte die Klassifikation zwischen den Klassen Gebäude und Fenster, Gebäude und Gebäudeteil-Mixtur sowie zwischen Fenster und Gebäudeteil-Mixtur werden, da hier die Hellinger-Distanzmatrix deutlich kleinere Werte als 0.1 hat.

Die vergleichsweise hohen Distanzen zur Trennung von Autos von anderen Klassen, liegt an der Ausprägung von Merkmal f_{10}, der Höhe des Mittelpunkts der Bounding Box einer Region. Entsprechende Verteilungen hatten wir in Abb. 4.1 gezeigt. Dort erkennt man, dass die Auto-Regionen nur kleine Werte annehmen, sich demnach in den Bildern eher im unteren Bereich befinden. Ab der Bildmitte gibt es im Prinzip keine Autos mehr in Bildern, also werden hier besonders hohe Differenzen bei der Bestimmung der Hellinger-Distanzen erzeugt, vgl. Gl. 4.2.

Die sehr niedrigen Hellinger-Distanzwerte für den Vergleich der klassenspezifischen Verteilungen für die Klassen Gebäude, Gebäudeteil-Mixtur und Fenster ist nicht überraschend. Die Annotationen von Instanzen dieser Klassen überlappen sich stark, somit ist auch eine schlechte Trennbarkeit für diese Klassen zu erwarten. Bei Inspektion der Distanz für die beiden Klassen Gebäude und Vegetation überrascht uns der kleine Wert ein wenig, der mit 0.158 nicht sehr hoch ist. Dabei ist auch dieser Wert verständlich: Wir betrachten den Datensatz `terra-1` mit den perspektivisch verzerrten Bildern. Durch die Verzerrungen könnten die gradienten- und texturbasierten Merkmale nicht sehr aussagekräftig sein. Vegetation ist meistens grün, d. h. die Regionen weisen hohe Intensitätswerte im grünen Kanal aus. Da viele Fassaden zudem hellgrau bis weiß angestrichen sind, können auch Gebäude-Regionen hohe Intensitätswerte im grünen Kanal haben. Eine Unterscheidung ist daher nur durch Kombination der Merkmale möglich.

4.2. Modellierung der Klassifikation

Im folgenden Abschnitt beschreiben wir unseren Versuchsaufbau für die Klassifikation der segmentierten Bildregionen. Wir führen dazu eine Kreuzvalidierung durch, bei der wir jede Region jedes Bildes genau einmal zum Testen verwenden. Für die Beurteilung des Klassifikationserfolgs unseres selbst entwickelten Klassifikationsverfahrens, den Alternierenden Entscheidungsbäumen mit K Klassen, verwenden wir maßstabsabhängige Projektionen in den LDA-Unterraum und eine MAP-Klassifikation mittels Gaußscher Mischverteilungen.

4.2.1. Durchführung der Klassifikation

Wir haben Regionen in diversen Maßstabsebenen segmentiert. In den oberen Maßstabsebenen zeigt eine Region oft eine komplette Fassade oder eine vollständige Baumkrone, während in den unteren Maßstabsebenen viele kleine Regionen vorkommen, die Ornamente an der Fassade oder eine Gruppe von Blättern eines Baumes darstellen. Daher teilen wir die Menge unserer Trainingsdaten in verschiedene Teilmengen auf und führen die Klassifikation maßstabsabhängig durch. Dazu bestimmen wir für jede stabile Region S_m die nächstgelegene Maßstabsebene, die einer Oktavstufe der Bildpyramide entspricht. Da wir unsere Wasserscheidenregionen in 41 Maßstabsebenen segmentieren, erhalten wir so eine Auswahl von fünf Maßstäben für die Klassifikation.

Die maßstabsspezifischen Verteilungen von vier ausgewählten Merkmalen zeigen wir in Abb. 4.10. V. l. n. r. zeigen wir die Verteilungen für die Höhe des Mittelpunkts der Bounding Box im Bild (Merkmal 10), die Differenz zwischen dem mittleren Blau-Wert der Region und ihrer Umgebung (Merkmal 21), die Differenz zwischen dem maximalen Wert des normalisierten Gradientenhistogramms und dem drittgrößten Wert (Merkmal 27) sowie die Orientierung der Hauptachse des generalisierten Vierecks (Merkmal 44). Dabei haben wir die Verteilungen um so dunkler dargestellt, je höher die Referenzebene im Maßstabsraum liegt. Wir erkennen in der Abbildung von Merkmal 10, dass die Regionen kleineren Maßstabs nahezu gleichverteilt bzgl. der Höhe im Bild liegen, die Merkmalsausprägungen in höheren Maßstabsebenen aber stärker oszillieren. Bei Merkmal 21 sind die Verteilungen niedriger Maßstabsebenen normalverteilt mit geringer Streuung, die bei höheren Maßstabsebenen deutlich größer ist. In der höchsten der fünf Maßstabsebenen würden wir die Verteilung sogar als Mischverteilung von zwei Gaußverteilungen modellieren. Ebenso können wir auch deutliche Unterschiede bei den Verteilungen der anderen beiden Merkmale feststellen.

Die Klassifikationsergebnisse evaluieren wir durch eine Kreuzvalidierung. Dazu teilen die Menge der Bilder in fünf gleichgroße Teilmengen, und in jedem Test verwenden wir die Regionen aus Bildern von vier Teilmengen zum Lernen des Klassifikators sowie die Regionen aus Bildern der übrigen Menge zum Testen. So stellen wir sicher, dass wir jede Region jedes Bildes genau einmal zum Testen verwenden. Da in den unteren Maßstabsebenen mehr Regionen segmentiert werden als in den oberen, variieren auch die Größen unserer Datensätze. Bei den Versuchen auf dem Benchmarkdatensatz, `terra-1`, haben wir Trainingsdaten von ca. 30 000 bis zu wenigen tausend Regionen pro Versuch.

In unseren ersten Experimenten haben wir mit allen K verschiedenen Klassen operiert, d. h. wir arbeiteten mit elf Klassen im ersten Datensatz `terra-1`, mit 38 Klassen im zweiten Datensatz `terra-2` sowie je 18 Klassen in den beiden anderen Datensätzen `terra-3` und `aerial`. Neben der hohen Zahl von Klassen kommt auch eine extrem ungleiche Häufigkeit der Klassen vor. So haben fast 32% der segmentieren Regionen auf Bildern der Benchmarkdaten aus `terra-1` das Klassenlabel `building`, aber nur etwas mehr als 1% das Klassenlabel `sky`.

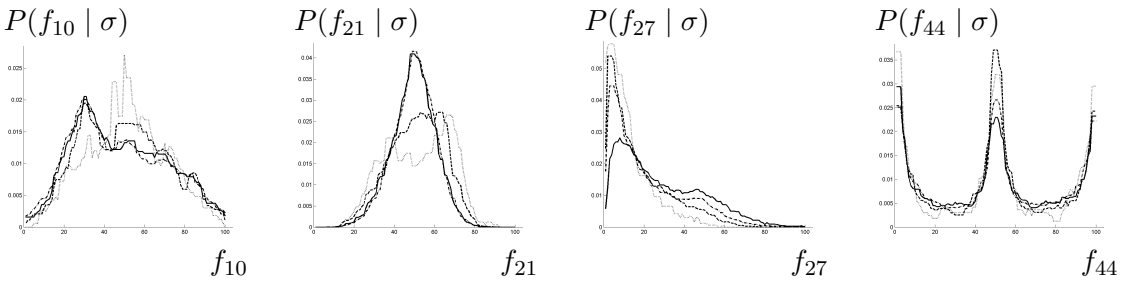


Abbildung 4.10.: In den Abbildungen zeigen wir die maßstabsspezifischen Verteilungen ausgewählter Merkmale. Die Verteilung eines Merkmals von Regionen aus Referenzebene 2 ist als durchgezogene Linie dargestellt, die aus Referenzebene 3 ist als gestrichelte Linie, die aus Referenzebene 4 ist als Strich-Punkt-Linie und die aus Referenzebene 5 ist als gepunktete Linie. Das Merkmal f_{10} stellt die Höhe des Mittelpunkts der Bounding Box im Bild dar, Merkmal f_{21} die Differenz zwischen dem mittleren Blau-Wert der Region und ihrer Umgebung. Des Weiteren zeigen wir mit Merkmal f_{27} die Differenz zwischen dem maximalen Wert des normalisierten Gradientenhistogramms und dem drittgrößten und mit Merkmal f_{44} die Orientierung der Hauptachse des generalisierten Vierecks.

Die relativen Häufigkeiten aller Klassen von `terra-1` haben wir nach Bestimmung des besten Klassenlabels $\widetilde{x}_m$ von allen 131 060 segmentierten Regionen bestimmt und führen sie in Tab. 4.2 auf.

Wir haben die Erfahrung gemacht, dass bei einer solchen Ungleichverteilung zwischen den Klassen starke Fehlklassifikationen auftreten, die vor allem die kaum vorkommenden Klassen betreffen. Selbst wenn wir eine ML-Klassifikation durchführen, bei der die A-priori-Wahrscheinlichkeiten für das Auftreten einer Klasse unberücksichtigt bleiben, haben wir diese starken Fehlklassifikationen. Gewichten wir unsere Trainingsdaten so, dass die marginalen Klassen stärkeren Einfluss haben, dann führt das zu starken Fehlklassifikationen bei den häufig vorkommenden Klassen.

Wir haben aus diesem Grund unsere Klassenzahl reduziert und verschiedene Klassen zusammengelegt. Dafür gibt es zwei Möglichkeiten. Einerseits können wir semantisch ähnliche Klasse zusammenfassen wie z. B. Bürgersteig und Fahrbahn zu Straße bzw. Boden oder Fensterscheiben zu Fenster zählen. Klassen wie Himmel oder Auto haben aber keine zu ihnen ähnlichen Klassen. Andererseits besteht die einfache Möglichkeit, alle Klassen mit einem Auftreten von unter einem festzulegenden Schwellwert zu einer neuen Klasse `others` zusammenzulegen. Diese neue Klasse ist dann sehr divers, bietet aber die Möglichkeit, eine weitere Klassifikation zur Bestimmung der ursprünglichen Klasse im Anschluss an die erste Klassifikation durchführen zu können. Wir haben uns für die letzte Variante entschieden, haben nur in wenigen Fällen noch manuell Klassen semantisch zusammengefügt, wie z. B. Fenster und Fensterscheiben in Datensatz `terra-2`. Auf eine weiterführende Klassifikation haben wir allerdings aus Zeitgründen verzichtet.

Alle Klassen mit einer relativen Häufigkeit von unter 3% haben wir zu der neuen Klasse zusammengefasst. Bzgl. des Benchmarkdatensatzes `terra-1` führt das zu einer Reduktion von elf auf sieben Klassen, wobei die ursprünglichen Klassen `pavement`, `road`, `sky` und `mixture` die neue Klasse `others` formen. Sie hat eine A-priori-Wahrscheinlichkeit von 9.12%.

Bei den anderen Datensätzen konnten wir so die Anzahl der Klassen von $\mathsf{K} = 38$ auf $K = 8$

Klasse	A-priori-Wahrscheinlichkeit
`building`	31.92%
`building-mixture`	3.03%
`car`	3.69%
`door`	1.09%
`mixture`	1.44%
`pavement`	2.76%
`road`	2.24%
`sky`	1.60%
`vegetation`	24.52%
`window`	23.71%
`background`	4.00%

Tabelle 4.2.: A-priori-Wahrscheinlichkeiten der Klassen von `terra-1`

(`terra-2`), von $\mathsf{K} = 18$ auf $K = 8$ (`terra-3`) sowie von $\mathsf{K} = 18$ auf $K = 6$ (`aerial`) reduzieren. Auffällig bei Datensatz `aerial` ist zudem, dass die Klasse `background` eine A-priori-Wahrscheinlichkeit von 77.25% hat, da sehr viele Bildbereiche wie Straßen und Vegetation nicht annotiert wurden.

4.2.2. Referenzklassifikation

In jedem Maßstab bestimmen wir mittels Linearer Diskriminanzanalyse (LDA) den Unterraum mit Dimension $K - 1$, in dem unsere K Klassen am besten trennbar sind (Duda *et al.*, 2001). In diesem LDA-Unterraum bestimmen wir klassenspezifische Wahrscheinlichkeitsverteilungen. Eine klassen- und maßstabsspezifische Analyse der Merkmale analog zur Analyse aller Merkmale in Kap. 4.1 haben wir nicht durchgeführt. Stattdessen approximieren wir jede dieser Verteilungen durch eine Mischverteilung dreier Gauß-Verteilungen im LDA-Unterraum. Die Mittelwerte dieser Gauß-Verteilungen erhalten wir durch den iterativen 3-Means-Ansatz. Die Ausgabe einer Wahrscheinlichkeitsverteilung wird anschließend noch mit der A-priori-Wahrscheinlichkeit für das Auftreten der Klasse verknüpft, um mittels MAP-Klassifikation bessere Ergebnisse zu erzielen.

Wenn wir einen Merkmalsvektor f_m einer stabilen Region S_m klassifizieren, dann projizieren wir ihn zuerst in den entsprechenden LDA-Unterraum und bestimmen dann die K vielen Wahrscheinlichkeiten für die Zugehörigkeit zur Klasse k, die wir normieren, um so die Wahrscheinlichkeiten $P(x_m^k)$ zu erhalten, vgl. Kap. 2.1. Das zu evaluierende Klassifikationsergebnis $\widehat{x}_m$ ist die Klasse mit der höchsten Wahrscheinlichkeit.

4.2.3. Ergebnisse

Ausführlicher stellen wir die Ergebnisse der Klassifikation der Regionen aus `terra-1` vor, die Ergebnisse der anderen drei Datensätze stellen wir im Anschluss etwas knapper dar.

Tab. 4.3 zeigt die Konfusionstabelle der Klassifikation mit sieben Klassen an, die wir im Datensatz `terra-1` verwenden. Jede Zeile zeigt die tatsächliche Klasse $\widetilde{x}_m$ der Regionen an, jede Spalte das Klassifikationsergebnis $\widehat{x}_m$. Die fetten Einträge auf der Hauptdiagonalen zeigen somit die richtigen Klassifikationen an. Die relativen Werte, d. h. wieviel Prozent der Regionen einer Klasse wurden entsprechend klassifiziert, zeigen wir ebenfalls in Tab. 4.3. Hier kann

Tabelle 4.3.: Referenzklassifikation des Datensatzes `terra-1`: Konfusionsmatrix und relative Erfolge der Klassifikation im LDA-Unterraum mit den $K = 7$ Klassen für Gebäude, Gebäudeteil-Mixturen, Autos, Vegetation, Fenster, Hintergrund und die eine zusammengelegte Klasse für den Rest und bei Verwendung von 131 060 Regionen. Die fett geschriebenen Werte zeigen die Detektionsraten der jeweiligen Klassen an. Jede Zeile zeigt die tatsächliche Klasse $\widetilde{x}_m$ der Regionen an, jede Spalte das Klassifikationsergebnis $\widehat{x}_m$.

Klasse	others	build.	build.-mix	car	veget.	window	backg.
others	**2 571**	3 315	42	688	2 636	2 275	422
	21.5%	27.7%	0.4%	5.8%	22.1%	19.0%	3.5%
building	1 589	**20 396**	176	471	7 179	11 714	303
	3.8%	**48.8**%	0.4%	1.1%	17.2%	28.0%	0.7%
building-mix	53	1 658	**61**	38	440	1 701	20
	1.3%	41.8%	**1.5**%	1.0%	11.1%	42.8%	0.5%
car	440	630	4	**1 154**	1 521	719	368
	9.1%	13.0%	0.1%	**23.9**%	31.4%	14.9%	7.6%
vegetation	478	3 022	22	458	**24 379**	3 321	452
	1.5%	9.4%	0.1%	1.4%	**75.9**%	10.3%	1.4%
window	409	7 822	112	349	3 781	**18 426**	174
	1.3%	25.1%	0.4%	1.1%	12.2%	**59.3**%	0.6%
background	299	740	2	527	2 588	767	**348**
	5.7%	14.0%	0.0%	10.0%	49.1%	14.6%	**6.6**%

man gut erkennen, dass ungefähr drei Viertel aller Vegetationsregionen, drei Fünftel aller Fensterregionen und fast die Hälfte aller Gebäuderegionen korrekt klassifiziert werden. Auf diesem Datensatz erzielen wir eine gesamte Erfolgrate für die Klassifikation im LDA-Raum von 51.4%.

Bei den anderen drei Datensätzen ähneln sich die Ergebnisse stark. Der Klassifikationserfolg bei Datensatz `terra-2` liegt bei nur 33.9% und fällt damit am schlechtesten aus. Beim Datensatz `terra-3` erzielen wir einen Klassifikationserfolg von 44.5% mit acht Klassen, und beim Luftbild-Datensatz haben wir einen Klassifikationserfolg von 48.1% mit sechs Klassen.

4.3. Alternierende Entscheidungsbäume für die Klassifikation mit mehreren Klassen

Im Folgenden stellen wir nun unser entwickeltes Klassifikationsverfahren vor: die Alternierenden Entscheidungsbäume (ADTboost) für die Klassifikation von Daten, die zu K verschiedenen Klassen gehören. Es gibt vor allem zwei Gründe für die Entwicklung des neuen Klassifikators. Erstens: Es gibt zwar sehr erfolgreiche andere Klassifkatoren wie Nächste Nachbarn (kNN), Support Vektor Maschinen (SVM) oder Random Forests (RF), aber sie haben dafür andere Nachteile, die zu einem Verzicht auf diese Methoden führen. Zweitens können wir das erfolgreiche Verfahren Adaboost durch eine hierarchische Anordnung der einfachen Klassifikatoren weiter verbessern.

Die beiden gravierenden Nachteile von kNN sind das hohe Speicheraufkommen sowie dessen

Ineffizienz, da es alle Trainingsdaten abspeichert und die Testdaten mit allen diesen Trainingsdaten vergleicht, um die nächstgelegenen Nachbarn zu finden. Selbst bei Verwendung einer effektiven Datenverwaltung ist dieses Klassifikationsverfahren zu langsam. SVMs scheiden nur aus, weil das Training des Klassifikators mit vielen stark überlappenden Klassen sehr langwierig ist. Für RFs haben wir zu wenige Trainingsdaten. Bei der Klassifikation auf Fassadenbildern verwenden z. B. Teboul *et al.* (2010) RFs aus mehreren Bäumen mit Tiefe 18. Vollständige Bäume mit Tiefe 18 haben $2^{17} = 131\,072$ Blätter. Wir haben im Datensatz `terra-1` insgesamt soviele Regionen, so dass wir bei dieser Tiefe keine aussagefähigen Verteilungen herleiten können. Kleinere Baumtiefen erzielen wiederum schlechtere Klassifikationserfolge.

Wie bereits in Kap. 2.3 geschildert, verknüpft ADTboost die Stärken von Adaboost mit den Stärken von Entscheidungsbäumen. Bei Adaboost werden viele einfache Klassifikatoren κ_t zu einem starken Klassifikator κ zusammensetzt. Der Erfolg von Adaboost wird noch größer, wenn man sehr effektive Klassifikatoren als einfache Klassifikatoren verwendet, dafür steigen die zeitlichen Kosten für das Training. Werden stattdessen die einfachen Klassifikatoren hierarchisch angeordnet, dann kann die Effektivität des zusammengesetzten Klassifikators verbessert werden, ohne dass dabei die zeitlichen Kosten für das Training explodieren.

Bevor wir den Mehrklassen-ADTboost herleiten, stellen wir kurz die Version mit zwei Klassen von Freund & Mason (1999) vor. Anschließend interpretieren wir die eingeführten Variablen und erläutern unsere Vorgehensweise zur Verallgemeinerung des ADTboost-Algorithmus.

4.3.1. ADTboost-Algorithmus

Freund & Mason (1999) haben Alternierende Entscheidungsbäume als binäre Klassifikationsmethode entwickelt, um das Problem des Overfittings herkömmlicher Entscheidungsbäume durch die Boosting-Formulierung zu reduzieren. In einfachen Entscheidungsbäumen werden in jedem inneren Knoten Entscheidungen getroffen, dabei entstandene Fehler können später nicht mehr behoben werden. Die Alternierenden Entscheidungsbäume enthalten zwar auch solche Entscheidungen, allerdings werden die Daten i. d. R. durch verschiedene sich nicht gegenseitig ausschließende Entscheidungsmöglichkeiten separiert, wobei kleinere Fehler kompensiert werden können.

Einige technische Details des Algorithmus sind in (Freund & Mason, 1999) fehlerhaft und wurden in (De Comité *et al.*, 2001) korrigiert. Das Verfahren hat folgende Eingabewerte: die Anzahl der durchzuführenden Iterationen T und M Datenpaare, bestehend aus einem D-dimensionalen Merkmalsvektor $\boldsymbol{f_m}$ und einem Klassenlabel $\widetilde{x}_m$. Der zusammengesetzte Klassifikator κ wird vom Algorithmus ausgegeben. Dieser besteht aus T einfachen Klassifikatoren κ_t und deren Gewichte ${\alpha_t}^+$ und ${\alpha_t}^-$. An die einfachen Klassifikatoren wird nur eine Bedingung geknüpft: Sie müssen besser klassifizieren als der Zufall. Wird kein solcher einfacher Klassifikator gefunden, dann wird die Suche nach weiteren einfachen Klassifikatoren beendet und die bisher zusammengestellte ADT-Baumstruktur ausgegeben. Diese hierarchische Struktur der einfachen Klassifikatoren formulieren wir durch die Vorbedingungen aus der Liste $\mathcal{P}$, die indirekt mit den einfachen Klassifikatoren κ_t verbunden ist. In der ursprünglichen Fassung von ADTboost gibt ein einfacher Klassifikator κ_t entweder eins der beiden Klassenlabel $+1$ bzw. -1 aus oder 0, wenn die Daten die Vorbedingung nicht erfüllen.

Im kombinierten Klassifikator κ gehen die Ergebnisse der einzelnen Klassifikatoren κ_t gewichtet ein, die entsprechenden Gewichte α_t spiegeln dabei den Erfolg der einzelnen Klassifikatoren wider. Freund & Mason (1999) haben die einfachen Klassifikatoren als Vergleich mit einem Schwellwert modelliert, d. h. der Merkmalsraum wird in $B = 2$ Bereiche unterteilt, und den beiden Bereichen wird je eine der beiden Klassen $+1$ und -1 zugeordnet. Die Klassifika-

Algorithm 4.1 Konzept des ADTboost-Algorithmus für zwei Klassen.

function ADTBOOST-2(T, $(\boldsymbol{f_m}, \widetilde{x}_m)_1^M$)
  ▷ T: Anzahl der Iterationen
  ▷ $(\boldsymbol{f_m}, \widetilde{x}_m)$: M Datenpaare mit $\widetilde{x}_m \in \{-1, +1\}$
  ▷ Ausgabe des zusammengesetzten Klassifikators $\kappa = (\kappa_t, {\alpha_t}^+, {\alpha_t}^-)_{t=1,\ldots,T}$
  Initialisiere Liste der Vorbedingungen $\mathcal{P}$
  Initialisiere Gewichte der Daten $w_1^m = \frac{1}{M}$
  ▷ gleiche Gewichtung der Daten zu Beginn
  Bestimme α_0
  for $t = 1, \ldots, T$ **do**
    Bestimme besten einfachen Klassifikator $\kappa_t : f \mapsto \{-1, 0, +1\}$
    ▷ Auswahl unter Berücksichtigung von $\mathcal{P}$ und w_t^m
    if Fehlerrate $\epsilon \geq 0.5$ **then**
      ▷ Einfacher Klassifikator ist nicht besser als Zufall.
      ▷ Ende von ADTboost.
      return κ.
    end if
    ▷ Einfacher Klassifikator wird akzeptiert.
    Bestimme ${\alpha_t}^+$ für Klasse $+1$
    Bestimme ${\alpha_t}^-$ für Klasse -1
    Aktualisiere die Liste der Vorbedingungen $\mathcal{P}$
    Bestimme neue Gewichtsverteilung w_{t+1}^m
  end for
  return κ.
end function

tion bewerten Freund & Mason (1999) in den beiden Bereichen getrennt: ${\alpha_t}^+$ gewichtet die Klassifikation von κ_t in Bezug auf den Bereich, in dem alle Daten zur Klasse $+1$ zugehörig klassifiziert werden, analog ${\alpha_t}^-$ im anderen Bereich bzgl. der Klasse -1. Das zusätzliche Gewicht α_0, das Freund & Mason (1999) zu Beginn des Algorithmus bestimmen, hängt von den vorhandenen Häufigkeiten der Klassen ab. Dabei gibt das Vorzeichen von α_0 Auskunft über die Klassenpräferenz, d. h. das Vorzeichen des Gewichts bekundet, welche der beiden Klassen $+1$ und -1 überwiegt, und der Betrag von α_0 drückt die Stärke dieser Klassenpräferenz aus. Daher interpretieren wir dieses Gewicht als a-priori-Klassifikation.

Die Formulierung des Algorithmus für ADTboost mit zwei Klassen in Alg. 4.1 haben wir relativ allgemein gehalten, so dass nur wenige Änderungen notwendig sind, um ein Konzept für die Mehrklassen-Klassifikation zu erhalten. In der Formulierung müssen wir nur drei Dinge verändern.

1. Ein einfacher Klassifikator unterscheidet nun K Klassen, die wir mit $1, 2, \ldots, K$ durchnummerieren.

2. Die a-priori-Gewichtung $\boldsymbol{\alpha}_0$ werden wir als einen K-dimensionalen Vektor definieren. Die Elemente von $\boldsymbol{\alpha}_0$ werden alle positiv sein und es gilt: Je größer das k-te Element im Verhältnis zu den anderen Elementen des Vektors ist, umso größer ist die Wahrscheinlichkeit für das Auftreten der k-ten Klasse.

3. Die einfachen Klassifikatoren müssen angepasst werden. Dazu führen wir noch einen weiteren Parameter B ein, der die Anzahl der Bereiche angibt, in die ein einfacher

Klassifikator den Merkmalsraum durch seine Entscheidungen zerlegt. Um die Daten in B disjunkte Bereiche für K Klassen einzuteilen, sind mindestens $B \geq \lceil \log_2 K \rceil$ viele binären Entscheidungen notwendig. Wenn wir jeden der B Bereiche individuell bewerten wollen, dann stellt $\boldsymbol{\alpha}_t$ nun einen B-dimensionalen Vektor von Gewichten dar. Bei der Kombination der einfachen Klassifikatoren zum starken Klassifikator κ werden die positiven Gewichte entsprechend der Ausgabe der einfachen Klassifikatoren kumuliert.

Diese Schritte für die Verallgemeinerung von ADTboost wurden bereits von Zhu *et al.* (2006) für einen K-Adaboost-Klassifikator vorgeschlagen und finden sich im Ansatz auch in (Viola & Jones, 2001a) wieder, wo die beiden Klassen durch 0 und 1 benannt werden. Den entsprechenden Algorithmus des Mehrklassen-ADTboost beschreiben wir in Alg. 4.2.

Algorithm 4.2 Konzept des ADTboost-Algorithmus für K Klassen.

function ADTBOOST(T,B, $(\boldsymbol{f_m}, \widetilde{x}_m)_1^M$)
  ▷ T: Anzahl der Iterationen
  ▷ B: Anzahl der Bereiche
  ▷ $(\boldsymbol{f_m}, \widetilde{x}_m)$: M Datenpaare mit $\widetilde{x}_m \in \{1, \ldots, K\}$
  ▷ Ausgabe des zusammengesetzten Klassifikators $\kappa = (\boldsymbol{\alpha}_0, (\kappa_t, \boldsymbol{\alpha}_t)_{t=1,\ldots,T})$
  Initialisiere Liste der Vorbedingungen $\mathcal{P}$
  Initialisiere Gewichte der Daten $w_1^m = \frac{1}{M}$
  ▷ gleiche Gewichtung der Daten zu Beginn
  Bestimme $\boldsymbol{\alpha}_0$
  for $t = 1, \ldots, T$ **do**
    Bestimme besten einfachen Klassifikator $\kappa_t : f \mapsto \{0, 1, \ldots, K\}$
    ▷ Auswahl unter Berücksichtigung von $\mathcal{P}$ und w_t^m
    if Fehlerrate $\epsilon \geq \frac{K-1}{K}$ **then**
      ▷ Einfacher Klassifikator ist nicht besser als Zufall.
      ▷ Ende von ADTboost.
      return κ.
    end if
    ▷ Einfacher Klassifikator wird akzeptiert.
    Bestimme $\boldsymbol{\alpha}_t$
    Aktualisiere die Liste der Vorbedingungen $\mathcal{P}$
    Bestimme neues Gewicht der Daten w_{t+1}^m
  end for
  return κ.
end function

Unseren Mehrklassen-Algorithmus in Alg. 4.2 haben wir direkt aus dem Zweiklassen-Algorithmus hergeleitet. Bei der Realisierung der einzelnen Schritte werden aber erst die komplizierten Details von ADTboost sichtbar. Die ADTboost-Formeln von Freund & Mason (1999) basieren auf den optimierten Formeln von Adaboost, die Schapire & Singer (1999) bewiesen haben. Bei der Erweiterung von ADTboost auf die Klassifikation von K Klassen haben wir versucht, diese originalen Formeln von Schapire & Singer (1999) und Freund & Mason (1999) weitestgehend beizubehalten und so zu verallgemeinern, dass wir für $B = 2$ und $K = 2$ die alten Formeln aus (Freund & Mason, 1999) bzw. (De Comité *et al.*, 2001) wiedererkennen können.

Einen Alternierenden Entscheidungsbaum bestehend aus vier einfachen Klassifikatoren zeigen wir in Abb. 4.11. Wir haben ihn fiktiv erstellt, da er nur zur Visualisierung der Kom-

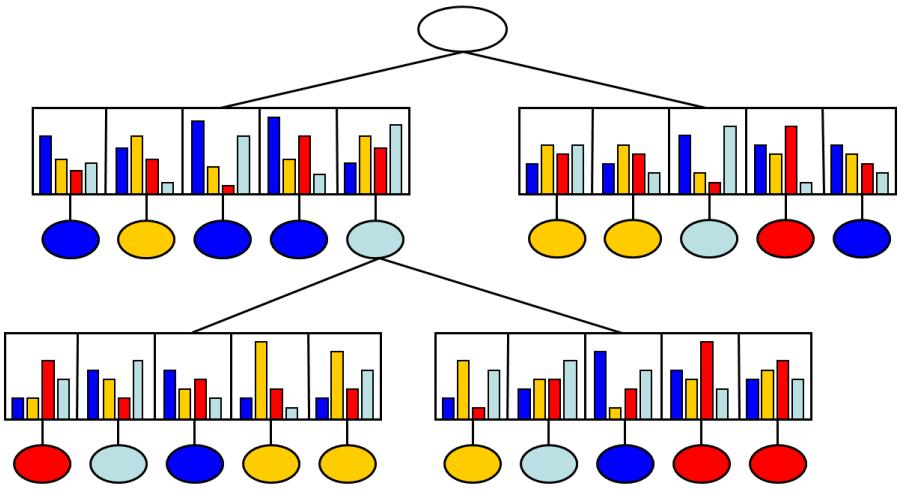


Abbildung 4.11.: Alternierender Entscheidungsbaum mit vier einfachen Klassifikatoren. Die länglichen rechteckigen Knoten repräsentieren die verschiedenen Klassifikatoren κ_t. Sie unterteilen (mittels vier Entscheidungen) den Merkmalsbereich in fünf Bereiche, die wir durch fünf Rechtecke kennzeichen. In jedem dieser Bereiche wird das Vorkommen der Klassen untersucht, die wir durch die senkrechten Balken in den Rechtecken visualisieren. Die Farbe gibt dabei die Klasse an, die Höhe eines Balkens korrelliert mit der Häufigkeit des Auftretens der entsprechenden Klasse. Zu jedem Bereich gehört ein Gewicht α_t, die durch die Ellipsen dargestellt werden. Da dieses Gewicht bei der Klassifikation für die stärkste Klasse votiert, haben wir die Farbe der Ellipsen entsprechend der Farbe des höchsten Balkens im Rechteck darüber gewählt. Der durch eine Ellipse repräsentierte Wurzelknoten stellt die a-priori-Gewichtung $\boldsymbol{\alpha}_0$ dar.

ponenten dient. Einen realen Alternierenden Entscheidungsbaum geben wir später bei der Demonstration des Verfahrens an. Freund & Mason (1999) haben die Klassifikationsmethode *Alternierenden Entscheidungsbaum* (ADT) genannt, weil sie einen Baum darstellt, bei dem jeder Pfad vom Wurzelknoten zu den Blättern abwechselnd rechteckige und elliptische Knoten passiert. Wir haben bei der Visualisierung dieser Klassifikationsmethode die Formen der Knoten übernommen.

4.3.2. Design der einfachen Klassifikatoren

In diesem Abschnitt stellen wir unsere einfachen Klassifikatoren κ_t vor, die in jeder Iteration t von ADTboost aus mehreren zur Verfügung stehenden Klassifikationskandidaten $\mathfrak{h}_h$ ausgewählt und in den bestehenden Alternierenden Entscheidungsbaum eingefügt werden. Dabei gehen wir noch nicht auf die Vorbedingungen ein, mit denen wir die Hierarchie der einfachen Klassifikatoren konstruieren.

Für die einfachen Klassifikatoren gibt es nach (Schapire & Singer, 1999) nur eine zu erfüllende Anforderung: Sie müssen besser klassifizieren als der Zufall. D. h. im Fall von zwei Klassen, dass die Fehlerrate eines einfachen Klassifikators ϵ kleiner als 0.5 sein muss, bei K Klassen unter $\frac{K-1}{K}$. Andere Bedingungen muss ein einfacher Klassifikator nicht erfüllen. Folglich wur-

den schon die verschiedensten Klassifikationsverfahren als einfache Klassifikatoren eingesetzt. Viola & Jones (2001a) verwenden bei ihrer Gesichtdetektion Kaskaden von Schwellwertvergleichen, die somit nicht balancierte Entscheidungsbäume darstellen. Entscheidungsbäume werden auch von Eibl & Pfeiffer (2005) als einfache Klassifikatoren vorgeschlagen. Weitere schwache Klassifikatoren können aber auch komplexere Klassifikationsverfahren wie Random Forests (RF) oder Support-Vektor-Maschinen (SVM) sein.

Wir haben eingangs unsere Gründe dargestellt, warum wir diese Verfahren nicht als generelle Klassifikatoren verwenden wollen. Für die RFs haben wir zu wenige Trainingsdaten, um RFs mit einer genügend großen Tiefe erzeugen zu können. Die SVMs scheiden wegen der hohen Laufzeit beim Training aus. Dazu verweisen wir auf ein Experiment von Rätsch *et al.* (2001), indem insgesamt 13 Datensätze klassifiziert wurden. Nach Angaben der Autoren mussten dazu ca. drei Millionen RBF-Netze und über 100 000 Teilprogramme berechnet werden. Unter Verwendung von 30 parallelverarbeitenden, leistungsstarken Computern trainieren Rätsch *et al.* (2001) ihre Klassifikatoren mehrere Tage lang, auf einem einzelnen Computer haben sie das Training mit einer Laufzeit von zwei Jahren abgeschätzt. In neueren Arbeiten geben (Rätsch & Warmuth, 2005) oder (Warmuth *et al.*, 2008) keine Details über die tatsächliche Laufzeit ihrer Verfahren an. Da heutige Rechner sehr viel leistungsfähiger sind als die von 2001, gehen wir auch von einer Beschleunigung des Lernens aus. Aber auch eine Laufzeit von nur mehreren Monaten ist für uns nicht akzeptabel.

Um effiziente, effektive und robuste Klassifikatoren κ_t zu verwenden, haben wir uns für Entscheidungsbäume entschieden, mit denen wir den Merkmalsraum unserer Daten in B Bereiche mit $B \geq K$ einteilen. Im Gegensatz zu den Bäumen von Eibl & Pfeiffer (2005) oder Shotton *et al.* (2008) verwenden wir lediglich ein Merkmal mit Index d bei deren Konstruktion. Diese strikte Wahl haben wir bereits in Kap. 2.3 mit der hohen Komplexität begründet. Außerdem können wir so auch den Einfluss von Merkmalen auf die Klassifikation belegen, vgl. dazu (Drauschke & Förstner, 2008). Um auch bei Verwendung des einen Merkmals möglichst aussagekräftige Entscheidungen zu finden, haben wir die Evaluation von Schwellwerten von Shotton *et al.* (2008) integriert. Sie wählen alle binären Entscheidungen des Baumes derart, dass der erwartete Gewinn an Information über die Verteilung der Klassen mit jeder Entscheidung maximiert wird (Lepetit *et al.*, 2005).

Shotton *et al.* (2008) konstruieren einen randomisierten binären Entscheidungsbaum, d. h. sie wählen für jede Entscheidung einige Merkmale und einige Schwellwerte zufällig aus und bestimmen dann die bestmögliche Entscheidung. Dabei werden diese Schwellwertkandidaten zufällig aus einer Gleichverteilung im Wertebereich des Merkmals gezogen. In unserem Fall haben wir viele normal- oder exponentialverteilte Merkmale, wo die zufallsbasierte Schwellwertsuche unter der Annahme von gleichverteilten Daten nicht zutrifft. Bevor wir nun den Zufallsgenerator manipulieren, haben wir uns für eine deterministische Vorgehensweise entschieden. Diese hat noch einen weiteren Vorteil: Wir können jede Entscheidiung erklären und daher auch überprüfen. Dazu bestimmen wir für jedes Merkmal $3(B-1)$ viele mögliche Schwellwerte, die wir bzgl. des Informationsgewinns evaluieren. Diese Kandidaten erhalten wir durch gleichverteilte Anordnung der Schwellwerte in der sortierten Liste der Merkmalsausprägungen. Dabei sortieren wir nicht alle Trainingsdaten, sondern bestimmen die entsprechenden Werte mit dem effizienten Suchalgorithmus von Hoare (1971). Die rekursive Schwellwertbestimmung endet, sobald wir die $B-1$ besten Schwellwerte gewählt haben. Da wir nur ein Merkmal nutzen, können diese Schwellwerte als Liste angeordnet werden, so dass der Merkmalsraum in B Intervalle eingeteilt wird. Die einzelnen Intervalle, unsere Bereiche, bezeichnen wir mit $\mathfrak{b}_b$.

Die Gewichtung der Daten beziehen wir bei der Bestimmung der Intervallgrenzen der B Bereiche nicht mit ein. Das hat den Vorteil, dass sich die Schwellwerte nicht ändern, wenn

die Gewichte der Daten w_t^m von Iteration zu Iteration kleiner oder größer werden. Das spart Rechenzeit bei der Auswahl des besten einfachen Klassifikators in jeder Iteration. Die Gewichte der Daten berücksichtigen wir aber bei der Klassifikation. Dazu bestimmen wir die klassenspezifischen Summen der Gewichte

$$W_k(b) = \sum w_t^m \quad \text{mit} \quad \widetilde{x}_m = \omega_k \text{ und } f_d \in \mathfrak{b}_b, \tag{4.4}$$

die wir als Likelihood-Werte interpretieren.

Die Klassifikation von neuen Daten $\boldsymbol{feat}$ funktioniert nun wie folgt: Zuerst schränken wir den Merkmalsraum auf das Merkmal f_d ein. Dann bestimmen wir durch Vergleich mit den Intervallgrenzen den Bereich $\mathfrak{b}_b$, in dem das Merkmal klassifiziert wird. Innerhalb des Bereichs kennen wir die Klasse mit dem höchsten Likelihood-Wert, die wir als Klassifikationsergebnis ausgeben. Dazu verwenden wir den Term ${\kappa_t}^b$ mit

$$\mathfrak{w}_b = \arg\max_k W_k(b). \tag{4.5}$$

Bei der Visualierung eines Alternierenden Entscheidungsbaums in Abb. 4.11 zeigen die rechteckigen Knoten des Baums die einfachen Klassifikatoren κ_t. Jedes dieser Rechtecke ist in fünf Teile unterteilt, die die fünf Bereiche $\mathfrak{b}_b$ darstellen. In jedem dieser Bereiche befinden sich vier farblich gekennzeichnete Balken, diese stellen die klassenspezifischen Summen der Gewichte $W_k(b)$ dar. Die Höhe des längsten Balkens spiegelt den Wert $\mathfrak{w}_b$ wieder, dessen Farbe wird vom darauf folgenden elliptisch geformten Knoten im Alternierenden Entscheidungsbaum übernommen, der das Gewicht der Klassifikation darstellt. Darauf gehen wir im folgenden Abschnitt näher ein.

4.3.3. Gewichtung der einfachen Klassifikatoren

Wir unterscheiden bei den klassifikationsentscheidenden Gewichten des Alternierenden Entscheidungsbaums zwischen seinem weißen Wurzelknoten und den farbigen Ellipsen weiter unten im Baum.

Bei den Gewichten $\boldsymbol{\alpha}_0$ handelt es sich um einen Vektor der Länge K, d. h. es gibt für jede Klasse ein Gewicht, das Auskunft über die Häufigkeit der Klasse gibt. Diese Elemente des Vektors, die Gewichte ${\alpha_0}^k$, werden vor der ersten Iteration bestimmt, wenn alle Daten gleich gewichtet werden. Die Gewichte der Daten w_1^m sind zu diesem Zeitpunkt gleich groß, d. h. die Daten werden gleichwertig beurteilt. Wir bestimmen die Gewichte durch

$${\alpha_0}^k = \frac{1}{2} \log \frac{\sum_i W_i + \varepsilon}{\sum_{i \neq k} W_i + \varepsilon}, \tag{4.6}$$

wobei W_k die Summe aller Gewichte der Daten sind, die zur k-ten Klasse gehören, d. h.

$$W_k = \sum w_1^m \text{ mit } \widetilde{x}_m = \omega_k. \tag{4.7}$$

Unter der Annahme, dass es sich die zu klassifizierenden Daten auf alle Klassen verteilen, ist der Bruch in Gl. 4.6 immer größer als 1 und dessen Logarithmus somit immer positiv. Die Variable ε ist eine kleine Zahl, die lediglich die Division durch 0 bzw. einen Logarithmus von 0 verhindern soll. Wir haben die Gl. 4.6 analog zur Gl. 4.8 konstruiert, so dass alle Gewichte ${\alpha_0}^k$ und ${\alpha_t}^b$ eine ähnliche Aussagekraft haben.

Bei den Gewichten $\boldsymbol{\alpha}_t$ handelt es sich um einen Vektor der Länge B, d. h. es gibt für jeden Bereich $\mathfrak{b}_b$ des Klassifikators ein Gewicht, das Auskunft über den Erfolg der Klassifikation in diesem Bereich gibt. Diese Elemente des Vektors, die Gewichte ${\alpha_t}^b$, berechnen wir durch:

$${\alpha_t}^b = \frac{1}{2} \log \frac{\sum_k W_k(b) + \varepsilon}{\sum_k W_k(b) - \mathfrak{w}_b + \varepsilon}. \tag{4.8}$$

Im Dividenden steht die Summe aller Gewichte von Daten, die bei der Klassifikation in den Bereich $\mathfrak{b}_b$ fallen. Da wir die klassenspezifischen Summen der Gewichte bereits vorliegen haben, bestimmen wir den Dividend als Summe der $W_k(b)$. Im Divisor werden ebenfalls alle Summen der Gewichte addiert, mit einer Ausnahme: es fehlt die Summe der Gewichte der Klasse, die der Klassifikator als beste ausgibt ($\mathfrak{w}_b$). Diese fehlende Summe ist wegen Gl. 4.5 die größte von allen. Da zudem alle Gewichte und damit auch ihre klassenspezifischen Summen nicht negativ sind, ist der Quotient definitiv größer als 1. Nach Anwendung des Logarithmus erhalten wir somit auf jeden Fall einen positiven Wert für ${\alpha_t}^b$. Klassifiziert κ_t im Bereich $\mathfrak{b}_b$ sehr gut, d. h. er gibt sehr wenige falsche Ergebnisse zurück, dann ist der Divisor sehr klein und ${\alpha_t}^b$ wird sehr groß. Daher spiegeln die Gewichte der einfachen Klassifikatoren deren Klassifikationserfolg wider.

4.3.4. Hierarchie der einfachen Klassifikatoren

In den vergangenen beiden Abschnitten haben wir die Knoten von Alternierenden Entscheidungsbäumen vorgestellt, nun gehen wir auf deren hierarchische Anordnung ein. Dazu thematisieren wir nun die Liste der Vorbedingungen $\mathcal{P}$. Die Elemente dieser Liste können wir als alle möglichen Pfade vom Wurzelknoten zu einem elliptischen Knoten des Baums visualisieren. Formal enthält die Liste $\mathcal{P}$ Prädikate $\mathfrak{p}$, d. h. Funktionen, die ein Merkmal f_d auf die beiden Wahrheitswerte 1 und 0 abbilden. Mit $\mathfrak{p}_t$ bezeichnen wir die Vorbedingung des Klassifikators κ_t, und die einfachen Klassifikatoren definieren wir nun mit integrierter Vorbedingung als

$$\kappa_t(m) = \left\{ \begin{array}{lll} k & \text{wenn} & \mathfrak{p}_t(m) = 1 \text{ und } f_d \in \mathfrak{b}_b \text{ und } \mathfrak{w}_b = W_k(b) \\ 0 & \text{wenn} & \mathfrak{p}_t(m) = 0 \end{array} \right\}. \tag{4.9}$$

Initialisiert wird die Liste der Vorbedingungen $\mathcal{P}$ mit dem Prädikat $\mathfrak{p}_0 = \mathtt{T}$, das immer 1 ausgibt und somit von allen Daten erfüllt wird (Freund & Mason, 1999). Der erste Klassifikator κ_1 kann nur an den bereits existierenden Wurzelknoten gehängt werden, d. h. seine Vorbedingung ist $\mathfrak{p}_1 = \mathfrak{p}_0$. Durch das Einfügen des Klassifikators in den Alternierenden Entscheidungsbaum nehmen wir B neue Ellipsen in den Baum auf. Deshalb entstehen B neue Vorbedingungen, die an die Bereiche $\mathfrak{b}_b$ des Klassifikators κ_t gekoppelt sind, d. h. $\mathcal{P}$ wird um B Elemente erweitert:

$$\mathcal{P} = \mathcal{P} \cup \left\{ \mathfrak{p}_t \wedge \mathfrak{p}^{t,b} \right\}_{b=1\ldots B} \text{ mit } \mathfrak{p}^{t,b} = \left\{ \begin{array}{lll} 1 & \text{wenn} & f_d \in \mathfrak{b}_b \\ 0 & \text{sonst} & \end{array} \right\}. \tag{4.10}$$

In Abb. 4.11 zeigen wir einen Alternierenden Entscheidungsbaum mit vier einfachen Klassifikatoren, die wir der Einfachheit wegen zeilenweise durchnummerieren. Dann haben die beiden oberen Klassifikatoren κ_1 und κ_2 nur die Vorbedingung $\mathfrak{p}_1 = \mathfrak{p}_2 = \mathtt{T}$. Die anderen beiden Klassifikatoren κ_3 (unten links) und κ_4 (unten rechts) hängen beide von einer Vorauswahl durch κ_1 ab. Die Vorbedingung von κ_3 und κ_4 ist dieselbe, weil beide einfachen Klassifikatoren an dieselbe Ellipse angehängt wurden. Die Vorbedingung lautet demnach $\mathfrak{p}_3 = \mathfrak{p}_4 = \mathtt{T} \wedge \mathfrak{p}^{1,5} = \mathfrak{p}^{1,5}$.

4.3.5. Auswahl der einfachen Klassifikatoren

Boosting-Verfahren verfolgen stets eine Greedy-Strategie: In jeder Iteration wird aus der Menge der möglichen Klassifikationskandidaten der Beste ausgewählt. Die Klassifikationskandidaten, die wir $\mathfrak{h}_h$ nennen, unterscheiden sich in unserem ADTboost-Verfahren in der Auswahl der Vorbedingung, d. h. in der Platzierung im Alternierenden Entscheidungsbaum, und im ausgewählten Merkmal. Wir diskutieren die Auswahl der einfachen Klassifikatoren κ_t bzw. die Bestimmung des besten Klassifikationskandidaten $\mathfrak{h}_h$ unter der Annahme, wir haben gleichgewichtete Daten. Auf die Rolle der Gewichtung gehen wir dann im nächsten Abschnitt ein.

Es hat sich herausgestellt, dass die Verallgemeinerung des Auswahlkriteriums komplizierter ist als die Verallgemeinerung der anderen Arbeitsschritte. Daher stellen wir zunächst das Auswahlkriterium nach Freund & Mason (1999) vor, das für den Fall $K = B = 2$ sinnvolle Ergebnisse liefert. Im Anschluss werden wir dieses Kriterium analysieren und daraus unsere Verallgemeinerung dieses Auswahlkriteriums herleiten.

Das Auswahlkriterium des besten Klassifikationskandidaten bei ADTboost ist eine Erweiterung des Adaboost-Auswahlkriteriums von Schapire & Singer (1999). Bei dem zu minimierenden Kriterium handelt es sich um die Funktion Z, die von Freund & Mason (1999) durch

$$Z(h) = 2 \sum_b \underbrace{\sqrt{W_+(\mathfrak{h}_h) W_-(\mathfrak{h}_h)}}_{Z_l} + W(\neg \mathfrak{h}_h), \tag{4.11}$$

definiert wurde. Auf den Wurzelterm gehen wir gleich noch genauer ein, daher bezeichnen wir diesen als linken Term der Gleichung Z_l. Der rechte Term, $W(\neg \mathfrak{h}_h)$, stellt die Summe der Gewichte aller Daten dar, die die Vorbedingung von $\mathfrak{h}_h$ nicht erfüllen. Im linken Term erscheinen die beiden Symbole W_+ und W_-, mit denen die Summe der Gewichte der Daten bezeichnet wird, die im Bereich $\mathfrak{b}_b$ des Kandidaten $\mathfrak{h}_h$ liegen und zur Klasse $+1$ bzw. -1 gehören. Da im Idealfall einer der beiden Bereiche die Klasse $+1$ zugewiesen bekommt und der andere Bereich die Klasse -1, interpretieren wir W_+ als die Summe der Gewichte der korrekt klassifizierten Daten und analog W_- als die Summe der Gewichte der falsch klassifizierten Daten in einem Bereich. Damit haben wir bereits den ersten Schritt zur Verallgemeinerung der Formel vollzogen.

Für die Analyse dieses Auswahlkriteriums haben wir einige Werte der Z-Funktion bestimmt und in Tab. 4.4 aufgelistet. Die Werte hängen (a) von der Fehlerrate ϵ_b des Klassifikationskandidaten im Bereich $\mathfrak{b}_b$ und (b) von der Auslesefähigkeit der Vorbedingung des Klassifikationskandidaten ab. Der Einfachheit halber nehmen wir gleiche Fehlerraten in allen Bereichen an. Auf die einzelnen Summanden Z_l gehen wir gleich ein. Bei balancierten Entscheidungen korreliert die Auslesekraft der Vorbedingung mit der Tiefe $\mathfrak{d}$ des Klassifikationskandidaten in der Hierarchie der Klassifikatoren. Dabei bedeutet $\mathfrak{d} = 1$, dass der Klassifikationskandidat direkt unter dem Wurzelknoten angebracht wird. Aus Tab. 4.4 erfahren wir, dass der beste Klassifikationskandidat derjenige ist, der ohne Fehler klassifiziert und dabei alle Daten berücksichtigt. Der zweitbeste ist dann der fehlerlose Klassifikationskandidat, der eine Vorbedingung enthält, die nur die Hälfte der Daten erfüllt. Der drittbeste Kandidat hat eine Fehlerrate von $\epsilon_b = 0.1$, aber keine einschränkende Vorbedingung, etc.

Als Analyse dieses Auswahlkriteriums fassen wir folgende Eigenschaften von Z zusammen:

1. Der linke Term aus Gl. 4.11 ist eine monoton steigende Funktion in Abhängigkeit von der Fehlerrate ϵ des Klassifikationskandidaten. Bei einer perfekten Klassikation mit $\epsilon = 0.0$ ist das geometrische Mittel der Gewichte $W_+ W_- = 0$ und damit der ganze linke Term. Als supremum für eine akzeptierbare Fehlerrate gilt der Fehler des Zufallsklassifikator

Tabelle 4.4.: Funktionswerte von Z berechnet nach Gl. 4.11. Z wurde berechnet für den Zweiklassenfall in Abhängigkeit von der Fehlerrate ϵ des Klassifikationskandidaten und seiner Tiefe im Baum $\mathfrak{d}$ unter der Annahme, dass alle Daten gleich gewichtet sind und alle Klassifikatoren die Datenmenge in zwei Hälften separieren.

	$\epsilon_b = 0.0$	$\epsilon_b = 0.1$	$\epsilon_b = 0.2$	$\epsilon_b = 0.3$	$\epsilon_b = 0.4$	$\epsilon_b = 0.5$
$\mathfrak{d} = 1$	0.000	0.600	0.800	0.916	0.979	1.000
$\mathfrak{d} = 2$	0.500	0.800	0.900	0.958	0.989	1.000
$\mathfrak{d} = 3$	0.750	0.900	0.950	0.979	0.994	1.000
$\mathfrak{d} = 4$	0.875	0.950	0.975	0.989	0.997	1.000

mit $\epsilon = 0.5$, die zu einem geometrischen Mittel von $W_+W_- = 0.25$ pro Bereich führt, wenn alle Daten klassifiziert werden. Der linke Term hat daher eine Obergrenze von 1.

2. Der rechte Term aus Gl. 4.11 ist nicht negativ und bestraft einen Klassifikationskandidaten, wenn nur ein Teil der Daten klassifiziert wird. Der rechte Term hat den Wert 0, wenn alle Daten bei der Klassifikation berücksichtigt werden. Er nimmt den Wert 0.5 an, wenn der Klassifikationskandidat einen Klassifikator als Vorbedingung hat, 0.75 bei zweien etc. Bezeichnen wir mit $\mathfrak{d}$ die Tiefe des Klassifikationskandidaten $\mathfrak{h}_h$ im Entscheidungsbaum der Klassifikatoren (also ohne Gewichtsknoten), dann ist $W(\neg\mathfrak{h}_h) = 1 - (\frac{1}{2})^{\mathfrak{d}-1}$. Dabei gilt für einen Klassifikationskandidaten mit Vorbedingung 1: $\mathfrak{d} = 1$.

3. Der linke Term aus Gl. 4.11 enthält eine Gewichtung von 2. Damit wird ein schlechter Klassifikationskandidat, der alle Daten klassifiziert, bestraft und ein guter Klassifikationskandidat kann einen niedrigeren Z-Wert auch dann erzielen, auch wenn er nur einen Teil der Daten berücksichtigt.

4. Unter den eingangs erwähnten Annahmen nimmt die Z-Funktion immer einen Wert zwischen 0 und 1 an. 0 bedeutet dabei, dass der Klassifikationskandidat alle Daten fehlerlos klassifiziert. Die 1 erreicht die Z-Funktion nur, wenn sie so schlecht wie der Zufall klassifiziert, egal ob auf allen Daten oder tiefer in der Hierarchie der Klassifikatoren.

Wir haben eben eine einfache Möglichkeit angedeutet, wie man das in Gl. 4.11 vorgestellte Auswahlkriterium auf den Mehrklassenfall abändern kann. Dazu lassen wir lediglich die Summe über B Bereiche laufen und interpretieren W_+ als die Summe der Gewichte der korrekt klassifizierten Daten im entsprechenden Bereich, analog W_-. Der rechte Term aus Gl. 4.11 bleibt auch im Mehrklassenfall unverändert und gibt die Summe der Gewichte der Daten an, die die Vorbedingung des Klassifikationskandidaten nicht erfüllen. Dass diese Adaption noch unzureichend ist, belegt eine Aufstellung der Z-Werte analog zu Tab. 4.4 mit $K = 4$ und $B = 4$, die wir in Tab. 4.5 dokumentieren. Dort stellen wir zwei Probleme fest: Erstens, die Z-Werte in einer Zeile steigen nicht monoton an, d. h. bei gleicher Vorbedingung wird ein Klassifikationskandidat mit hoher Fehlerrate, z. B. $\epsilon = 0.75$, besser bewertet als ein Klassifikationskandidat mit mittlerer Fehlerquote, z. B. $\epsilon = 0.45$. Zweitens, die Gewichte der Daten, die die Vorbedingung nicht erfüllen, sind so groß, dass selbst ein perfekter Klassifikationskandidat in der zweiten Hierarchieebene keine Chance hat, besser bewertet zu werden als ein zufällig klassifizierender Klassifikationskandidat, der alle Daten berücksichtigt.

Dass der linke Term aus Gl. 4.11 bei wachsender Fehlerrate für $K > 2$ nicht monoton steigt, liegt an der Mittelung der beiden Summen W_+ und W_- mittels einer Quadratwurzel.

Tabelle 4.5.: Funktionswerte von Z berechnet nach einer direkten Verallgemeinerung von Gl. 4.11. Z wurde berechnet für den Mehrklassenfall mit $K = B = 4$ in Abhängigkeit von der Fehlerrate ϵ des Klassifikationskandidaten und seiner Tiefe im Baum $\mathfrak{d}$ unter der Annahme, dass alle Daten gleich gewichtet sind und alle Klassifikatoren die Datenmenge in vier gleiche Viertel separieren.

	$\epsilon_b = 0.0$	$\epsilon_b = 0.15$	$\epsilon_b = 0.3$	$\epsilon_b = 0.45$	$\epsilon_b = 0.6$	$\epsilon_b = 0.75$
$\mathfrak{d} = 1$	0.000	0.357	0.458	0.497	0.489	0.433
$\mathfrak{d} = 2$	0.750	0.839	0.864	0.874	0.872	0.858

In Abb. 4.12 zeigen wir u. a. den Graph der Funktion Z_l als durchgezogene Linie mit

$$Z_l = \sqrt{\frac{W_+}{W_+ + W_-}\frac{W_-}{W_+ + W_-}} = \sqrt{(1-\epsilon_b)\epsilon_b}. \tag{4.12}$$

Dazu normieren wir die Summen der Gewichte der korrekt bzw. falsch klassifizierten Daten, so dass wir die Summanden im linken Teil der Z-Funktion als Z_l in Abhängigkeit der Fehlerquote ϵ_b formulieren. Da Klassifikationskandidaten $\mathfrak{h}_h$ besser klassifizieren sollen als der Zufall, kann Z_l bei $K = 2$ nur Werte links der Maximalstelle von $\epsilon = 0.5$ annehmen, ist demnach monoton steigend im Intervall $\epsilon_b \in [0, 0.5]$. Für $K > 2$ ändert sich die obere Grenze für die akzeptierte Fehlerquote, aber der Graph von Z_l (definiert nach Gl. 4.12) steigt nicht mehr monoton an. Die Berechnung des Klassifikationserfolgs Z_l sollte so modifiziert werden, dass Z_l ihr Maximum erreicht, wenn die Fehlerquote des Zufallsklassifikators erreicht wird, d. h. bei $\epsilon_b = \frac{K-1}{K}$.

In Abb. 4.12 haben wir auch zwei weitere Graphen dargestellt, die einen monotonen Anstieg bis zur Maximalstelle $\epsilon_b = \frac{K-1}{K}$ haben und durch folgende Berechnungsvorschrift definiert werden:

$$Z_l = \left(W_+(\mathfrak{h}_h)\right)^{1/K} \left(W_-(\mathfrak{h}_h)\right)^{(K-1)/K}. \tag{4.13}$$

Gl. 4.13 erfüllt unsere Anforderung, im Intervall $\epsilon = \left[0, \frac{K-1}{K}\right]$ monoton steigende Werte für Z_l zu liefern. Zudem gilt für $K = 2$, dass Gl. 4.13 den Klassifikationserfolg gleich bewertet wie Gl. 4.12. So haben wir eine Verallgemeinerung des Zweiklassenfalls auf den Mehrklassenfall bewirkt.

Das zweite Problem haben wir dabei noch nicht gelöst, denn noch immer gilt: Die Werte von $W(\neg\mathfrak{h}_h)$ sind zu groß. Ein perfekter Klassifikationskandidat in der zweiten Hierarchieebene hat nach wie vor keine Chance, besser bewertet zu werden als ein zufällig klassifizierender Klassifikationskandidat der ersten Hierarchiebene. Dies zeigt Tab. 4.6, wo die Berechnung des Klassifikationserfolgs durch Gl. 4.13 erfolgt, der Rest aus Gl. 4.11 noch unverändert ist.

Das Problem mit der Bewertung von Klassifikationskandidaten aus unterschiedlichen Hierarchiestufen des (Alternierenden) Entscheidungsbaums können wir nur durch eine neue Gewichtung der beiden Terme aus Gl. 4.11 erzielen. Dazu normieren wir den linken bzw. den rechten Term derart, dass wir deren maximal erreichbaren Werte im Mehrklassenfall auf die maximal erreichbaren Werte im Zweiklassenfall normieren. Wer bestimmen unser Auswahlkriterium durch

$$Z(h) = \left(\frac{B}{2}\right)^{\mathfrak{d}} \frac{1}{\frac{(K-1)^{(K-1)/K}}{K}} \left(\sum_b W_+(\mathfrak{h}_h)^{1/K} W_-(\mathfrak{h}_h)^{(K-1)/K}\right) + \frac{1-\left(\frac{1}{2}\right)^{\mathfrak{d}-1}}{1-\left(\frac{1}{B}\right)^{\mathfrak{d}-1}} W(\neg\mathfrak{h}_h), \tag{4.14}$$

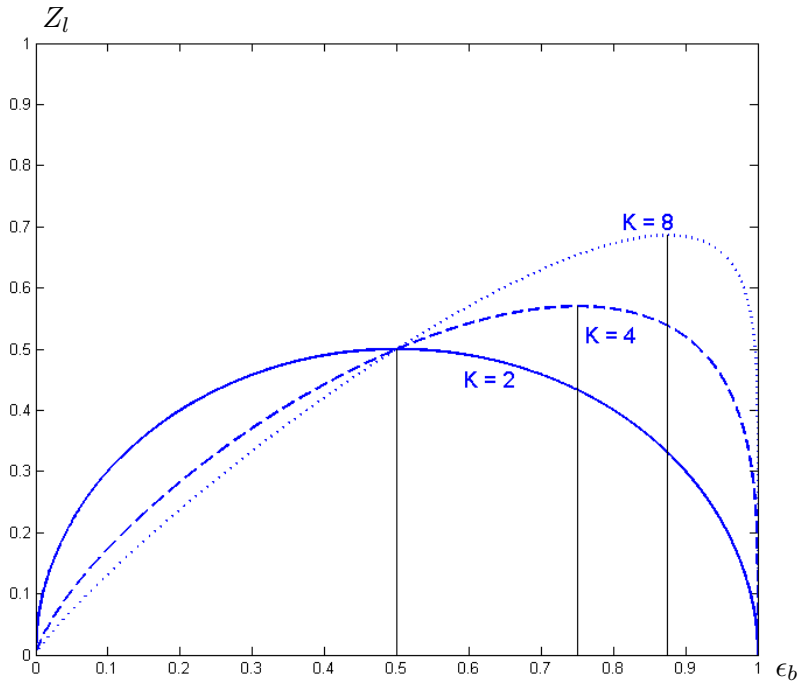


Abbildung 4.12.: Bestimmung des Klassifikationserfolgs Z_l in Anhängigkeit von der Fehlerrate ϵ und mit Parameter K, der Anzahl der Klassen.

Tabelle 4.6.: Funktionswerte von Z berechnet nach der Verallgemeinerung von Gl. 4.11 unter Verwendung von Gl. 4.13. Z wurde berechnet für den Mehrklassenfall mit $K = B = 4$ in Abhängigkeit von der Fehlerrate ϵ des Klassifikationskandidaten und seiner Tiefe im Baum $\mathfrak{d}$ unter der Annahme, dass alle Daten gleich gewichtet sind und alle Klassifikatoren die Datenmenge in vier gleiche Viertel separieren.

	$\epsilon_b = 0.0$	$\epsilon_b = 0.15$	$\epsilon_b = 0.3$	$\epsilon_b = 0.45$	$\epsilon_b = 0.6$	$\epsilon_b = 0.75$
$\mathfrak{d} = 1$	0.000	0.231	0.370	0.473	0.542	0.569
$\mathfrak{d} = 2$	0.750	0.807	0.842	0.868	0.885	0.892

Tabelle 4.7.: Funktionswerte von Z berechnet nach Gl. 4.14. Z wurde berechnet für den Mehrklassenfall mit $K = B = 4$ in Abhängigkeit von der Fehlerrate ϵ des Klassifikationskandidaten und seiner Tiefe im Baum $\mathfrak{d}$ unter der Annahme, dass alle Daten gleich gewichtet sind und alle Klassifikatoren die Datenmenge in vier gleiche Viertel separieren.

	$\epsilon = 0.0$	$\epsilon = 0.15$	$\epsilon = 0.3$	$\epsilon = 0.45$	$\epsilon = 0.6$	$\epsilon = 0.75$
$\mathfrak{d} = 1$	0.000	0.406	0.650	0.830	0.951	1.000
$\mathfrak{d} = 2$	0.500	0.703	0.825	0.915	0.975	1.000
$\mathfrak{d} = 1$	0.750	0.851	0.912	0.957	0.987	1.000
$\mathfrak{d} = 2$	0.875	0.925	0.956	0.978	0.993	1.000

d. h. unsere Bewertung hängt sowohl von der Anzahl der Klassen K als auch von der Anzahl der Bereiche B ab. Für den Fall $K = B = 2$ erzielen wir dieselben Z-Werte wie in Tab. 4.4 dokumentiert. Für den Fall $K = B = 4$ nimmt unserer Auswahlkriterium nun Werte an, die unsere Beobachtungen aus dem Zweiklassenfall erfüllen. Insbesondere gilt, dass Klassifikationskandidaten mit einer niedrigeren Fehlerquote bzw. mit einer schwächeren Vorbedingung bevorzugt werden.

4.3.6. Aktualisierung der Gewichte der Daten

Die bisherigen Schritte des ADTboost-Algorithmus haben wir so diskutiert, als ob die Gewichte der Daten w_t^m gleich groß sind. Durch die Gewichtung der Daten kann man den Fokus auf diejenigen Daten lenken, die falsch klassifiziert wurden, und den Einfluss von den Daten reduzieren, die korrekt klassifiziert wurden. Diese Gewichte wirken bei der Erstellung der einfachen Klassifikationskandidaten sowie bei der Auswahl und Beurteilung des besten einfachen Klassifikators κ_t mit. Die Neugewichtung der Daten erfolgt immer im letzten Schritt einer Iteration, siehe Alg. 4.2.

Schapire & Singer (1999) benutzen das Gewicht α_t des ausgewählten einfachen Klassifikators κ_t zur Aktualisierung der w_t^m, weil der Betrag von α_t stark mit dem Klassifikationserfolg zusammenhängt. Freund & Mason (1999) haben die Aktualisierung von Schapire & Singer (1999) weitestgehend übernommen und nur dahingehend angepasst, dass sie mit zwei Gewichten ${\alpha_t}^+$ und ${\alpha_t}^-$ die Klassifikation bewerten:

$$w_{t+1}^m = \left\{ \begin{array}{lll} w_t^m e^{\widetilde{x}_m {\alpha_t}^+} & \text{wenn} & \kappa_t(m) = +1 \\ w_t^m e^{-\widetilde{x}_m {\alpha_t}^-} & \text{wenn} & \kappa_t(m) = -1 \\ w_t^m & \text{wenn} & \kappa_t(m) = 0 \end{array} \right\} \cdot , \tag{4.15}$$

anschließend werden die Gewichte so normiert, dass $\sum w_{t+1}^m = 1$ ist.

Wichtig hierbei ist in Gl. 4.15 vor allem, dass sich die Gewichte jener Daten nicht bzw. nur leicht durch die Normierung ändern, die die Vorbedingung des ausgewählten einfachen Klassifikators nicht erfüllen. Bei der Mehrklassifikation mit Zerlegung des Merkmalsraums in B Bereiche, wird der Anteil dieser Daten erheblich größer. Wir haben dabei festgestellt, dass die Änderung der Gewichte in Gl. 4.15 bewirkt, dass sich die Gewichte der falsch klassifizierten Daten stark ändern können und dann die einfachen Klassifikatoren nur noch tief im Alternierenden Entscheidungsbaum eingefügt werden. Dieses Overfitting kann verhindert werden,

indem auch die Gewichte der Daten vergrößert werden, die die Vorbedingung nicht erfüllen. Wir verändern daher die Gewichte der Daten durch

$$w_{t+1}^m = \left\{ \begin{array}{lll} w_t^m e^{-\alpha_t{}^b} & \text{wenn} & f_d \in \mathfrak{b}_b \ \wedge \ \kappa_t(m) = \widetilde{x}_m \\ w_{t+1}^m e^{\alpha_t{}^b} & \text{wenn} & f_d \in \mathfrak{b}_b \ \wedge \ \kappa_t(m) \neq \widetilde{x}_m \\ w_{t+1}^m e^{W(\neg \mathfrak{p}_t} & \text{wenn} & \kappa_t(m) = 0 \end{array} \right\}, \tag{4.16}$$

wobei $W(\neg \mathfrak{p}_t$ die Summe der Geichte der Daten darstellt, die die Vorbedingung $\mathfrak{p}_t$ des einfachen Klassifikators κ_t nicht erfüllen. Anschließend normieren wir die Gewichte wieder (Freund & Mason, 1999).

4.3.7. Zusammengesetzter Klassifikator

Nun haben wir alle Komponenten von ADTboost erörtert, es fehlt nur noch die Zusammensetzung der einfachen Klassifikatoren zum stärkeren Klassifikator κ. Dies erledigen wir in zwei Schritten. Zuerst bestimmen wir die kumulativen Gewichte ρ_k für jede Klasse, danach wird die Klasse mit dem höchsten kumulativen Gewicht als Klassifikationsergebnis ausgegeben.

Das kumulative Gewicht ρ_k der k-ten Klasse berechnen wir durch

$$\rho_k(m) = \alpha_0(k) + \sum_{t=1}^{T} \sum_{b=1}^{B} \alpha_t{}^b(m) \mathbb{I}(f_d \in \mathfrak{b}_b \ \wedge \ \kappa_t(\mathrm{m}) = \mathrm{k}), \tag{4.17}$$

wobei $\mathbb{I}$ eine Indikatorfunktion darstellt, die 1 zurückgibt, wenn ihr Argument wahr ist, sonst 0. Wenn wir die kumulativen Gewichte normieren, dann erhalten wir Wahrscheinlichkeiten β_k für die Klassenzugehörigkeiten von Daten mit

$$\beta_k(m) = \frac{\rho_k(m)}{\sum_i \rho_i(m)}, \tag{4.18}$$

die wir in unser bedingtes Bayes-Netz integrieren können.

Das Klassifikationsergebnis für die Evaluation der Klassifikation mit Alternierenden Entscheidungsbäumen erhalten wir durch Auswahl der Klasse mit dem höchsten kumulativen Gewicht:

$$\kappa(m) = \arg\max_k \rho_k(m). \tag{4.19}$$

4.3.8. Demonstration von ADTboost auf synthetischen Daten

Um die Funktionsweise von ADTboost nach der Präsentation der einzelnen Schritte in den vorangegangenen Abschnitten zu illustrieren, demonstrieren wir den Ablauf der Klassifikation an einem kleinen synthetisch generierten Beispieldatensatz. Er ist eine Erweiterung des 2-Klassen-Datensatzes, den wir bereits in (Drauschke & Förstner, 2008) zu Demonstrationszwecken verwendet haben. Wir zeigen die Daten in Abb. 4.3.8. Die drei Klassen können wir durch achsenparallele Geradenstücke perfekt trennen, d. h. dieser Datensatz sollte sehr gut durch ADTboost mit den von uns gewählten einfachen Klassifikatoren klassifizierbar sein.

Wir haben die Daten aus einer Gleichverteilung über dem Intervall $[0, 1]^2$ gewonnen und anschließend die Klassenzugehörigkeit bestimmt. Das Verhältnis der Klassen untereinander ist in etwa $2 : 1 : 1$, in der Reihenfolge schwarz, rot und blau. Vor der ersten Iteration wird die A-priori-Gewichtung $\boldsymbol{\alpha}_0$ nach Gl. 4.6 bestimmt. Wir erhalten für die schwarze Klasse einen Wert von 0.5239, für die rote Klasse einen Wert von 0.1989 und für die blaue Klasse den Wert

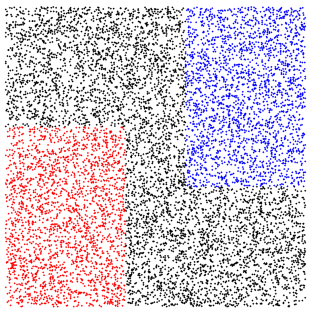


Abbildung 4.13.: Synthetisch generierter Datensatz mit zwei Merkmalen und drei Klassen. Die Daten haben wir zufällig aus einer Gleichverteilung auf dem Intervall $[0,1]^2$ bestimmt. Anschließend haben wir die Zugehörigkeit der Daten zu den drei Klassen festgelegt. Die rote Klasse enthält alle Daten mit $f_1 < 0.4 \wedge f_2 < 0.6$, die blaue Klasse besteht aus denjenigen Daten mit $f_1 > 0.6 \wedge f_2 > 0.4$, und alle anderen Daten gehören zur schwarzen Klasse.

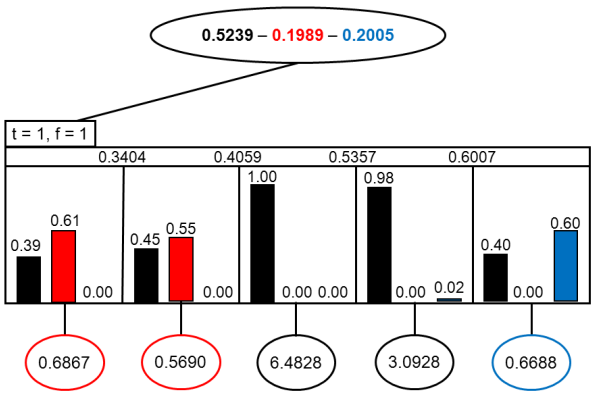


Abbildung 4.14.: Alternierender Entscheidungsbaum nach der ersten Iteration. Der Klassifikator κ besteht zu diesem Zeitpunkt aus dem Gewicht $\boldsymbol{\alpha}_0$, dem ersten einfachen Klassifikator κ_1, der den Merkmalsraum in fünf Bereiche einteilt, und fünf Gewichten $\boldsymbol{\alpha}_1$ für die Bewertung der Klassifikation in diesen Bereichen.

0.2005. Die beiden kleinen Klassen werden in $\boldsymbol{\alpha}_0$ ungefähr gleich bewertet, die größere schwarze Klasse hat ein Gewicht, das etwas mehr als das Doppelte des Gewichts der anderen Klassen ist. Durch die Gewichtung nach Gl. 4.6 werden demnach größere Klassen etwas bevorzugt.

In der ersten Iteration kann an den einzigen bestehenden Knoten des Klassifikationsbaums, dem Wurzelknoten $\boldsymbol{\alpha}_0$, ein Klassifikator angefügt werden, siehe Abb. 4.14. Die Klassifikation auf dem ersten Merkmal wird besser beurteilt als die auf dem zweiten Merkmal. Bei der Zerlegung des Merkmalsraums in fünf Bereiche werden 0.3404, 0.4059, 0.5357 und 0.6007 als bestmögliche Schwellwerte ausgwählt. Die optimalen Grenzen wären 0.4 und 0.6, die beinah perfekt gefunden wurden. Bei der anschließenden Klassifikation der fünf Bereiche nach der überwiegenden Klasse werden die ersten beiden Bereiche zur roten Klasse zugehörig markiert, die beiden mittleren Bereiche haben vor allem Daten der schwarzen Klasse, im letzten Bereich überwiegt die blaue Klasse. Als Gewichte des Klassifikators $\boldsymbol{\alpha}_1$ bestimmen wir nach Gl. 4.8 Werte zwischen 0.5690 und 6.4828. Die kleinen Werte unter 1 zeigen eine verhältnismäßig schlechte Klassifikation mit einer hohen Fehlerrate an, die hohen Werte kommen dort vor, wo (fast) keine falsch klassifizierten Daten vorliegen.

Als letzter Schritt in der ersten Iteration aktualisieren wir die Gewichte der Daten: Die

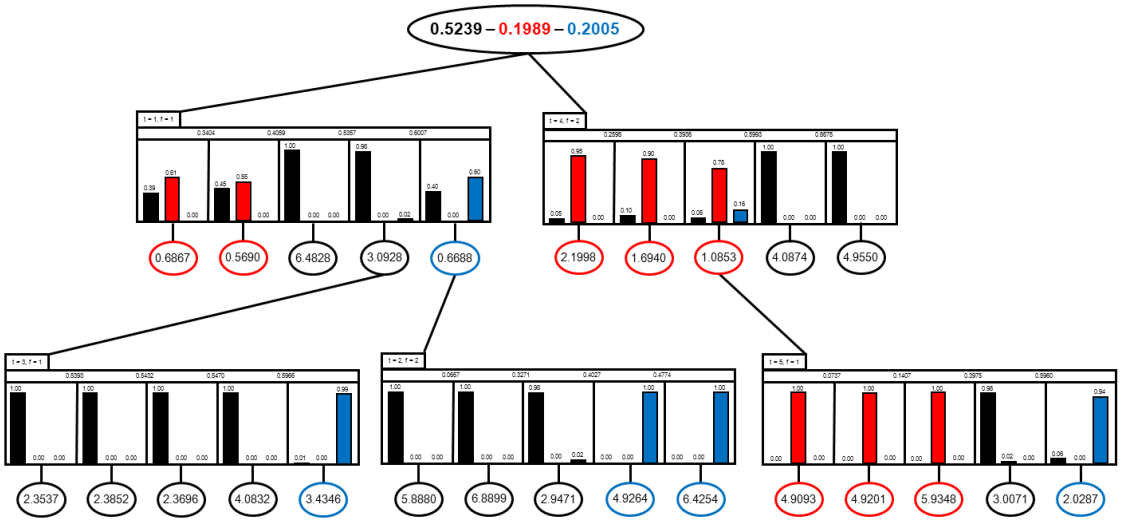


Abbildung 4.15.: Alternierender Entscheidungsbaum nach fünf Iteration.

Gewichte aller zur roten Klasse gehörenden Daten werden leicht gesenkt, weil sie von einem schwachen Klassifikator korrekt klassifiziert wurden. Analog sinken auch die Gewichte fast aller Daten, die zur blauen Klasse gehören. Die Gewichte der Daten der schwarzen Klasse mit $0.4059 < f_1 < 0.6007$ senken wir sehr stark, weil diese ein guter Klassifikator korrekt klassifiziert hat. Dagegen werden die übrigen Daten der schwarzen Klasse, d. h. die Punkte oberhalb der roten Klasse bzw. unterhalb der blauen Klasse in Abb. 4.3.8, höher gewichtet. So wundert es uns nicht, dass der zweite einfache Klassifikator κ_2 unterhalb des fünften Bereichs des ersten Klassifikators eingefügt wird. Dieser neue Klassifikator berücksichtigt κ_2 etwa 40% der Daten und hat auf diesen Daten eine sehr gute Performanz, wie die Verteilungen innerhalb der rechteckigen Bereiche darstellen, siehe Abb. 4.15. Folglich erhält er hohe Gewichte $\boldsymbol{\alpha}_2$ für seine Klassifikation.

In den weiteren Iterationen werden der dritte Klassifikator κ_3 unterhalb des vierten Bereichs des ersten Klassifikators, der vierte Klassifikator κ_4 wieder direkt an den Wurzelknoten und der fünfte Klassifikator κ_5 unterhalb des dritten Bereichs des vierten Klassifikators gehängt. Der dritte Klassifikator hat eine sehr einschränkende Vorbedingung, die nur 6.67% aller Daten erfüllen. An diesem Beispiel wird deutlich, dass der Alternierende Entscheidungsbaum zu Overfitting neigt, wenn die einfachen Klassifikatoren unter entsprechender Vorbedingung sehr gute Resultate liefern. Den Alternierenden Entscheidungsbaum nach fünf Iterationen zeigen wir in Abb. 4.15. Da der Klassifikationsbaum schnell eine unübersichtliche Größe erreicht, haben wir die wichtigen Details der Abbildung farblich illustriert. In den fünf Bereichen zeigen Balkendiagramme die jeweiligen Verteilungen der drei Klassen an, die zugehörigen Ellipsen haben wir entsprechend des Klassifikationsresultats farblich umrandet.

Nun zeigen wir die Vorgehensweise der Klassifikation mit Alternierenden Entscheidungsbäumen anhand von drei Beispielen. Der Datenpunkt mit dem Merkmalsvektor $[0.5, 0.5]$ liegt in der Mitte des Merkmalsraums und gehört zur schwarzen Klasse, die wir mit dem Klassenlabel 1 bezeichnen. Nach Gl. 2.10 setzen wir den starken Klassifikator κ aus den vielen einfachen

Abbildung 4.16.: Visualisierung der Klassifikationsergebnisse nach t Iterationen der synthetisch generierten 2D-Daten.

Klassifikatoren κ_t zusammen und bestimmen aus den kumulativen Gewichten ρ_k

$$\begin{array}{rcl} \rho_k\left([0.5, 0.5]\right) & = & \alpha_0(k) + \sum_{t=1}^{T} \sum_{b=1}^{B} \alpha_t{}^b \mathbb{I}([0.5, 0.5] \in \mathfrak{b}_b \wedge \kappa_t\left([0.5, 0.5]\right) = \mathrm{k}) \\ \rho_1\left([0.5, 0.5]\right) & = & 0.5239 + 6.4828 + 3.0071 = 10.0138 \\ \rho_2\left([0.5, 0.5]\right) & = & 0.1989 + 1.0853 = 1.2842 \\ \rho_3\left([0.5, 0.5]\right) & = & 0.2005 \end{array} \tag{4.20}$$

das beste Klassifikationsergebnis

$$\begin{array}{rcl} \kappa\left([0.5, 0.5]\right) & = & \arg\max_k \rho_k\left([0.5, 0.5]\right) \\ & = & \arg\max_k \left[10.0138, 1.2842, 0.2005\right] \\ & = & 1. \end{array} \tag{4.21}$$

Für die Weiterverwendung im Bayes-Netz benötigen wir die Wahrscheinlichkeiten, wie zuversichtlich der kombinierte Klassifikator κ bei seiner Entscheidung für die Klasse k ist. Dazu setzen wir die verschiedenen kumulativen Gewichte ρ_k ins Verhältnis zu deren Summe und erhalten:

$$\begin{array}{rcl} p(\widehat{x} \mid [0.5, 0.5]) & = & \frac{\rho_k([0.5,0.5])}{\sum_i \rho_i([0.5,0.5])} \\ & = & \frac{[10.0138,1.2842,0.2005]}{11.4985} = [0.8709, 0.1117, 0.0174] \end{array} \tag{4.22}$$

Wir visualisieren die jeweiligen Klassifikationsergebnisse nach den ersten fünf Iterationen in Abb. 4.16. Man erkennt gut, dass die erste Klassifikation eine Streifeneinteilung des Merkmalsraums bewirkt. In der folgenden Iteration wird ein einfacher Klassifikator konstruiert, der die korrekte Klassifikation der Ecke unten rechts bewirkt. Durch den vierten einfachen Klassifikator kann auch die obere linke Ecke korrekt klassifiziert werden. Bei den anderen beiden Klassifikatoren werden nur die bisherigen Abgrenzungen zwischen Klassen leicht verschoben. In den weiteren Iterationen ändern sich in der Ausgabe des Klassifikationsergebnisses praktisch nichts mehr, weshalb auf eine Darstellung der Klassifikationsergebnisse nach T Iterationen verzichten.

4.3.9. Klassifikationsergebnisse von ADTboost auf Benchmark-Daten

Wir schließen dieses Unterkapitel mit der Präsentation von weiteren Ergebnissen ab, die wir bei der Klassifikation mit ADTboost auf bekannten Datensätzen erzielen.

Als ersten Datensatz verwenden wir Daten aus vier acht-dimensionalen Normalverteilungen, deren Parameter Fukunaga (1972) angibt. Wir haben für jede dieser vier Klassen 1000 Daten zufällig bestimmt, die wir in Abb. 4.17 visualisieren. Dabei zeigen wir immer zwei verschiedene

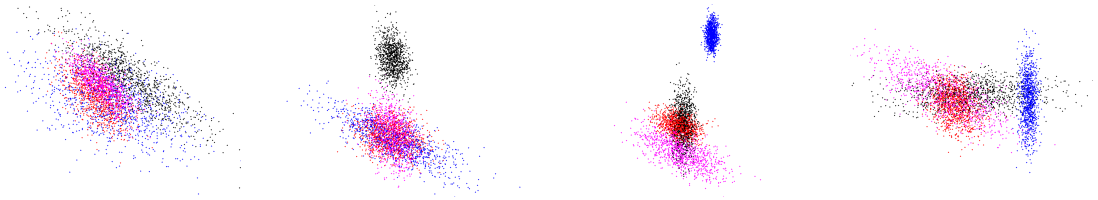


Abbildung 4.17.: Visualisierung des Datensatzes von Fukunaga (1972). V. l. n. r. zeigen wir jeweils zwei verschiedene Merkmale des 8D-Merkmalsraums, u. z. die Merkmale f_1 und f_2 in im linken Plot, f_3 und f_4 rechts daneben etc.

Tabelle 4.8.: Statistische Angaben zu Benchmark-Datensätzen: Anzahl der Klassen K, Anzahl der Merkmale D, Anzahl der Trainingsdaten pro Versuch, Anzahl der Testdaten pro Versuch, Anzahl der Versuche.

Datensatz	K	D	# Train.	# Testd.	# Versuche
`8D-Gauss`	4	8	1000	3000	100
`dermatology`	6	34	90	268	100
`glass`	6	10	57	157	100
`ionosphere`	2	34	89	262	100
`iris`	3	4	39	111	100
`sat`	6	36	4435	2000	1
`segmentation`	7	19	210	2100	1
`zoo`	7	16	28	73	100

Merkmale des 8D-Merkmalsraums. Die Parameter für die Beschreibung der klassenspezifischen Verteilungen, Mittelwerte und Kovarianzmatrizen, führen wir in Anhang A.2 in den Gl. A.1 bis A.5 auf. Die Klassen überlappen teilweise stark, die schwarz und blau dargestellten Klassen sind aber gut von den anderen separierbar. Wir bezeichnen diesen Datensatz mit `8D-Gauss`.

Neben dem Datensatz von Fukunaga haben wir ADTboost auch auf ein paar Benchmark-Datensätzen, die im Repository der UCI erhältlich sind, siehe (Asuncion & Newman, 2007). Diese Datenbank enthält eine Vielzahl von Datensätzen, von denen wir insgesamt sieben ausgewählt haben. Zwei davon haben bereits definierte Trainings- und Testdaten, die fünf anderen sind relativ kleine Datensätze, was ein effiziente Trainieren der Klassifikatoren für eine Kreuzvalidierung mit 100 Versuchen ermöglicht.

Die verwendeten Datensätze und ihre Eigenschaften haben wir in Tab. 4.8 zusammengestellt. Wir haben sie unverändert verwendet, mit Ausnahme des Datensatzes `dermatology`, bei dem wir acht Merkmale aus dem Originaldatensatz entfernt haben, weil diese nicht für alle Daten angegeben wurden. Bei unseren Experimenten haben wir eine Kreuzvalidierung mit i. d. R. 100 Versuchen durchgeführt. In Tab. 4.8 geben wir zusätzlich noch die Anzahl der verwendeten Trainings- und Testdaten je Versuch an. Bei den beiden Datensätzen `sat` und `segmentation` haben wir nur einen Versuch durchgeführt, weil hier der Benchmark die Einteilung in Trainings- und Testdaten bereits enthält.

Wir haben die in Tab. 4.8 aufgeführten Datensätze unter Verwendung der folgenden Verfahren klassifiziert. Auf Random Forests haben wir bei dem Vergleich verzichtet, weil alle Datensätze bis auf `sat` nur sehr wenige Trainingsdaten enthalten.

- **Adaboost und ADTboost**. Wir haben den Mehrklassen-Adaboost-Algorithmus von

Tabelle 4.9.: Fehlerraten bei der Klassifikation der Benchmark Datensätze

Datensatz	Adaboost	ADTboost	ML1	ML4	5NN	SVM	PER
`8D-Gauss`	4.5%	5.6%	13.0%	11.1%	0.9%	1.2%	n.a.
`dermatology`	23.5%	39.3%	n.a.	n.a.	26.2%	68.6%	8.9%
`glass`	19.7%	10.3%	82.1%	66.9%	5.5%	33.5%	6.8%
`ionosphere`	12.8%	18.1%	n.a.	n.a.	18.5%	n.a.	17.4%
`iris`	9.8%	7.0%	n.a.	n.a.	4.7%	5.1%	10.2%
`sat`	40.2%	37.2%	48.1%	56.5%	30.2%	77.0%	n.a.
`segmentation`	16.2%	13.0%	n.a.	n.a.	16.7%	75.9%	8.4%
`zoo`	23.7%	20.3%	n.a.	n.a.	18.2%	7.3%	10.3%

Zhu *et al.* (2006) implementiert und mit unserem ADTboost-Verfahren verglichen. Bei beiden Methoden haben wir allerdings andere einfache Klassifikatoren verwendet als oben vorgestellt. D. h. wir bestimmen zwar immer noch $B - 1$ viele Schwellwerte, haben aber bei diesem Experiment die Schwellwerte gleichverteilt im Raum eines Merkmals zwischen dem Minimal- und dem Maximalwert bestimmt. Die oben dargestellte Auswahl der Schwellwerte nach Shotton *et al.* (2008) haben wir erst später in unseren ADTboost-Algorithmus integriert. Wir halten diese Veränderung für marginal und haben daher auf eine Wiederholung der Experimente mit den hier vorgeschlagenen einfachen Klassifikatoren verzichtet. Bei beiden Verfahren haben wir das Boosting mit $T = 100$ Iterationen durchgeführt.

- **Maximum Likelihood Klassifikation**. Hier approximieren wir die jeweiligen Klassen durch eine bzw. als Mischverteilung von vier Normalverteilungen (ML1 bzw. ML4). Bei den Datensätzen, wo wir weniger Trainingsdaten in mindestens einer Klasse hatten als Merkmale, haben wir diese beiden Klassifikationsverfahren nicht durchgeführt, da wir hier die Verteilungen nicht lernen konnten.

- **STPRtool**. In der Matlab-Toolbox STPR von Franc & Hlaváč (2004) wurden einige Klassifikationsverfahren implementiert, von denen wir drei ebenfalls in unseren Experimenten verwendet haben. Diese sind die Klassifikation mittels der fünf nächsten Nachbarn (5NN), Support Vektormaschinen (SVM) und ein Linearer Klassifikator auf der Basis des Perceptron-Algorithmus (PER). Als Parameter haben wir bei der 5NN-Klassifikation das euklidische Distanzmaß und bei den SVMs die Verwendung von RBF-Kernels für die Modellierung der Trennfunktion eingestellt. Wir haben in drei Fällen die Experimente abgebrochen, weil Matlab entweder ein Speicherplatzproblem hatte oder weil das Lernen des Klassifikators eines Tests den Zeitraum von zwei Wochen überschritt.

Die von uns erzielten gemittelten Fehlerraten bei der Klassifikation der Testdaten präsentieren wir in Tab. 4.9. Wenn in der Tabelle ein **n.a.** (nicht anwendbar) steht, dann konnten wir die Klassifikation an diesem Datensatz mit der entsprechenden Methode nicht durchführen.

Im Vergleich mit den anderen Klassifikationsverfahren schneidet ADTboost auf den acht Benchmark-Datensätzen zufriedenstellend ab. ADTboost ist in der Lage, für jeden Datensatz Ergebnisse auszugeben, was neben den beiden Boosting-Verfahren sonst nur 5NN gelingt. Bzgl. 5NN ist aber die Performanz des Klassifikators zu kritisieren: Bei den vergleichsweise großen Datensätzen `8D-Gauss`, `sat` und `segmentation` haben wir bereits die Dauer der ineffizienten Vergleiche mit den Trainingsdaten gespürt. Die beiden ML-Verfahren haben große Probleme mit der geringen Anzahl von Trainingsdaten und wurden daher in sechs von acht Fällen

abgebrochen. Auch SVM und PER haben in ein bis zwei Fällen versagt, d. h. das Lernen des Klassifikators hat zu lange gedauert.

Die besten Klassifikationsergebnisse liefert 5NN und PER. 5NN klassifiziert immer dann korrekt, wenn die Klassen Cluster bilden und sich nicht stark überlappen. Lineare Klassifikatoren wie PER können bei den fünf kleinen Datensätzen `dermatology`, `glass`, `ionosphere`, `iris` und `zoo` gut abschneiden, weil die Kombination aus wenigen Trainingsdaten und hochdimensionalen Merkmalsräumen lineare Trennfunktionen für die effektive Separierung der Klassen ermöglicht. Daher hatten wir auch bessere Ergebnisse für die beiden Boosting-Verfahren erwartet, allerdings tendierenen sie wegen der kleinen Datensätze bei vergleichsweise vielen Iterationen ($T = 100$) zu Overfitting.

Wenn Adaboost und ADTboost ähnliche Ergebnisse liefern, dann bringt die hierarchische Anordnung der Klassifikatoren keine bessere Klassifikation. Bzgl. des Datensatzes `8D-Gauss` erkennen wir in den Plots von Abb. 4.17, dass sich mindestens zwei Klassen gut ohne hierarchische Anordnung der linearen einfachen Klassifikatoren separieren lassen. Mit den Ergebnissen von ADTboost auf den Datensätzen `dermatology` und `ionosphere` sind wir unzufrieden. Da auch 5NN sehr schlechte Ergebnisse liefert, überlappen sich die Klassen einander stark. Daraus schließen wir, dass die von uns gewählten einfachen Klassifikatoren κ_t keine guten Schwellwerte für die Einteilung des Merkmalsraums in B Bereiche finden können.

4.4. Experimente

Wir schließen dieses Kapitel mit einer Beurteilung der ADTboost-Klassifikation von Regionen unter Verwendung ihrer Merkmale ab. Zu Beginn gehen wir kurz auf die Wahl der zu lernenden Parameter ein. Dann präsentieren wir statistische Angaben über den Klassifikationserfolg unseres ADTboost-Verfahrens zur Klassifikation der segmentierten stabilen Regionen des Datensatzes `terra-1`. Anschließend visualisieren wir die Ergebnisse, d. h. wir zeigen Bilder mit den klassifizierten Regionen. In beiden Fällen gehen wir nur kurz auf die Ergebnisse der anderen drei Datensätze ein. Zum Schluss beurteilen wir die Klassifikation mit Alternierenden Entscheidungsbäumen.

4.4.1. Bestimmung der Parameter von ADTboost

Wir haben zwei Parameter von ADTboost zu bestimmen, alle anderen Werte werden aus den Daten abgeleitet. Bei den beiden Parametern handelt es sich um die Anzahl der Iterationen des Verfahrens T, die auch der Anzahl der einfachen Klassifikatoren im ADT entspricht, und um die Anzahl der Bereiche B, in die ein Klassifikator κ_t den Merkmalsraum eines Merkmals zerlegt.

Bei unseren Experimenten zur Klassifikation der Regionen aus Gebäudebildern haben wir Folgendes beobachtet: Die Anzahl B und die Tiefe des Alternierenden Entscheidungsbaums stehen in einem indirekten Verhältnis. Wird B groß gewählt, entstehen kleine Unterteilungen des Merkmalsraums und die hierarchische Struktur der einfachen Klassifikatoren verflacht. Andererseits können die einfachen Klassifikatoren bessere Klassifikationsergebnisse liefern, wenn sich die Klassen stark überlappen und der Merkmalsraum in kleine Intervalle zerlegt wird. Die Laufzeit zur Suche des besten einfachen Klassifikationskandidaten hängt von B und K ab, so dass ADTboost effizienter arbeitet, wenn B klein gewählt wird. Um allen Klassen eine Chance geben zu können, haben wir $B = K$ gesetzt. Größere Werte von B führten eher zu einem Klassifikator mit Overfitting, so dass wir auf eine kleinräumige Zerlegung des Merkmalsraums verzichtet haben.

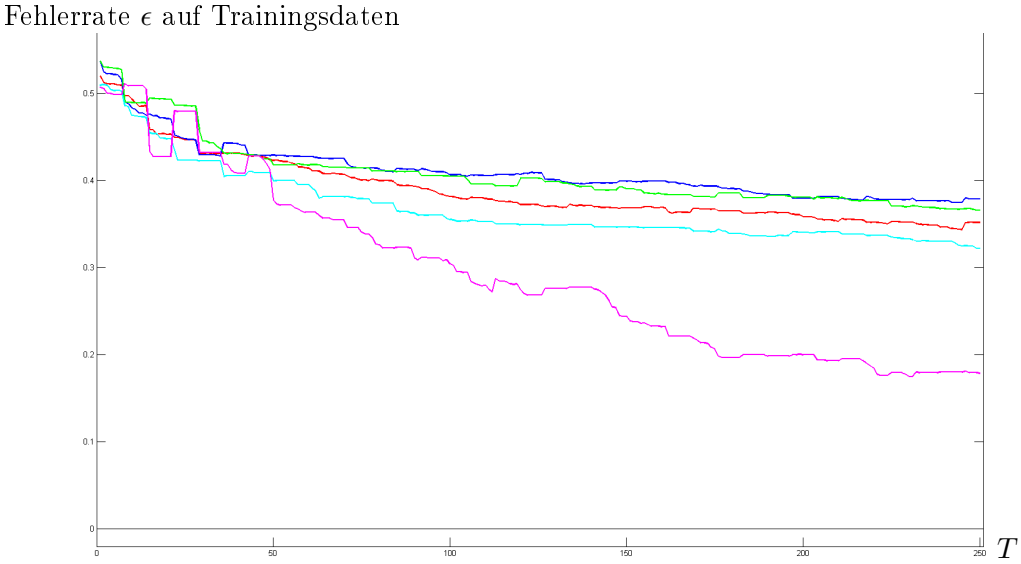


Abbildung 4.18.: Entwicklung der Fehlerrate ϵ bei steigender Anzahl der Iterationen T. Der magenta-farbene Graph zeigt die Fehlerrate ϵ bei der Klassifikationen der Regionen der höchsten Referenz-Maßstabsebene, die auf unter 0.2 sinkt. Die Fehlerraten bzgl. der anderen vier Maßstäbe (in blau,grün, rot und türkis) entwickeln sich sehr ähnlich und sinken von anfangs $\epsilon > 0.6$ auf $\epsilon < 0.4$.

Es gibt bislang keine analytische Methode zur Bestimmung der Anzahl der Iterationen T. So haben wir in Experimenten die Klassifikationsergebnisse in Abhängigkeit von T, aber auch von B untersucht. Wir konnten feststellen, dass die Klassifikation immer besser wird, je mehr einfache Klassifikatoren wir verwendet haben. In Abb. 4.18 haben wir die Fehlerraten ϵ auf den Trainingsdaten der Fassadenaufnahmen dargestellt, getrennt nach den fünf Referenz-Maßstäben. Die Fehlerraten ϵ nehmen zwar nicht monoton ab bei steigendem T, aber die Tendenz ist dennoch eine Verringerung der Fehler. Besonders erfolgreich sind wir bei der Klassifikation von Regionen mit der höchsten Referenzskala, die eine Fehlerrate von unter 20% auf den Trainingsdaten erzielen. In den anderen Maßstabsebenen erzielen wir Fehlerraten von lediglich 30 bis 40%. Im nächsten Abschnitt präsentieren wir die Klassifikationsergebnisse der Testdaten.

4.4.2. Statistische Auswertung

Tab. 4.10 zeigt die Konfusionstabelle der Klassifikation von Datensatz `terra-1` mittels Alternierenden Entscheidungsbäumen mit sieben Klassen. Jede Zeile zeigt die tatsächliche Klasse $\widetilde{x}_m$ der Regionen an, jede Spalte das Klassifikationsergebnis $\widehat{x}_m$. Die fetten Einträge auf der Hauptdiagonalen zeigen somit die richtigen Klassifikationen an. Die relativen Werte, d. h. wieviel Prozent der Regionen einer Klasse entsprechend klassifiziert wurden, zeigen wir direkt unterhalb der absoluten Zahlen. Hier kann man gut erkennen, dass ungefähr fast 70% aller Gebäude- und Vegetationsregionen sowie mehr als die Hälfte aller Fensterregionen korrekt klassifiziert werden. Auf dem vollständigen Datensatz erzielen wir einen Klassifikationserfolg von 54.9%.

Im Vergleich zwischen der Klassifikation mit Alternierenden Entscheidungsbäumen und unserer Referenzklassifikation im LDA-Unterraum können wir eine leichte Verbesserung des

Tabelle 4.10.: Konfusionsmatrix und relative Erfolge der Klassifikation mittels ADTboost mit den $K = 7$ Klassen für Gebäude, Gebäudeteil-Mixturen, Autos, Vegetation, Fenster, Hintergrund und die eine zusammengelegte Klasse für den Rest und bei Verwendung von 131 060 Regionen. Die fett geschriebenen Werte auf der Hauptdiagonalen zeigen die Erkennungsraten der jeweiligen Klassen an. Jede Zeile zeigt die tatsächliche Klasse $\widetilde{x}_m$ der Regionen an, jede Spalte das Klassifikationsergebnis $\widehat{x}_m$.

Klasse	others	build.	build.-mix	car	veget.	window	backg.
others	**3 291**	5 710	38	231	1 351	1 206	122
	27.6%	47.8%	0.3%	1.9%	11.3%	10.1%	1.0%
building	893	**28 768**	214	76	4 023	7 722	132
	2.1%	**68.8**%	0.5%	0.2%	9.6%	18.5%	0.3%
building-mix	44	2 275	**39**	7	217	1 381	8
	1.1%	57.2%	**1.0**%	0.2%	5.5%	34.8%	0.2%
car	497	2 263	28	**267**	829	846	106
	10.3%	46.8%	0.6%	**5.5**%	17.1%	17.5%	2.2%
vegetation	570	6 676	54	97	**22 597**	1 951	187
	1.8%	20.7%	0.2%	0.3%	**70.3**%	6.1%	0.6%
window	190	12 075	179	58	1 714	**16 831**	26
	0.6%	38.8%	0.6%	0.2%	5.5%	**54.2**%	0.1%
background	376	2 672	7	63	1 602	444	**107**
	7.1%	50.7%	0.1%	1.2%	30.4%	8.4%	**2.1**%

Tabelle 4.11.: Differenzen der Klassifikationsraten bzgl. des Datensatzes `terra-1`. Wir geben die Klassifikationserfolge von ADTboost im Vergleich mit der MAP-Klassifikation im LDA-Unterraum an.

Klasse	others	build.	build.-mix	car	veget.	window	backg.
others	+**6.1**%	+20.1%	−0.1%	−3.9%	−10.8%	−8.9%	−2.5%
building	−1.7%	+**20.0**%	+0.1%	−0.9%	−7.6%	−9.5%	−0.4%
building-mix	−0.2%	+15.4%	−**0.5**%	−0.8%	−5.6%	−8.0%	−0.3%
car	+1.2%	+33.8%	+0.5%	−**18.4**%	−14.3%	+2.6%	−5.4%
vegetation	+0.3%	+11.3%	+0.1%	−1.1%	−**5.6**%	−4.2%	−0.8%
window	−0.7%	+13.7%	+0.2%	−0.9%	−6.7%	−**5.1**%	−0.5%
background	+1.4%	+36.7%	+0.1%	−8.8%	−18.7%	−6.2%	−**4.5**%

Klassifikationserfolges feststellen, die von 51.5% auf 54.9% steigt. In Tab. 4.11 zeigen wir die Unterschiede zwischen den Ergebnissen der beiden Klassifikationsverfahren. Die Verbesserung der Klassifikation erklären wir mit der deutlich besseren Erkennungsrate von Gebäuderegionen mit ADTboost. Die Erkennungsraten der anderen Klassen nehmen ab, mit Ausnahme der Klasse `others`, deren Erkennungsrate leicht steigt. Generell stellen wir fest, dass die Ausgabe des Klassifikationsergebnisses `building` stark zunimmt: Bei der Klassifikation im LDA-Unterraum werden 37 583 Regionen als Gebäude klassifiziert, wovon 20 396 auch Gebäude sind (54.3%). Bei der Klassifikation mit ADTboost werden dagegen 60 439 Regionen als Gebäude bestimmt, wovon 28 768 auch Gebäude sind (47.6%).

Auch bei den anderen drei Datensätzen erzielen wir Ergebnisse bei der Klassifikation mit ADTboost, die vergleichbar mit den Ergebnissen bei der Klassifikation im LDA-Unterraum sind. Der Klassifikationserfolg bei Datensatz `terra-2` beträgt bei acht verwendeten Klassen 33.6%, was eine Veränderung von −0.3% gegenüber dem Klassifikationserfolg im LDA-Unterraum darstellt. Beim Datensatz `terra-3` mit ebenfalls acht Klassen erzielen wir einen Klassifikationserfolg von 42.0% (−2.5%), und beim Luftbild-Datensatz mit sechs Klassen haben wir einen Klassifikationserfolg von 51.4% (+3.3%).

4.4.3. Visualisierung der Ergebnisse

Neben der Statistik wollen wir nun die Ergebnisse der Klassifikation von Regionen mittels ADTboost visuell präsentieren. Dazu bieten sich Klassifikationskarten an, wie eingangs im Konzept formuliert, siehe Abb. 2.1. Darin stellen wir alle Regionen in gelb dar, die ADTboost als Instanzen der entsprechenden Klasse erkannt hat.

In einer ersten Präsentation visualisieren wir die Ergebnisse nach Klassen und Maßstabsebene getrennt. Dazu zeigen wir in Abb. 4.19 die Regionen, die ADTboost als Instanzen der Klassen `building` (oben), `window` (Mitte) und `vegetation` (unten) erkannt hat. Das dazugehörige Originalbild ist Teil des Benchmark-Datensatzes `terra-1` und wurde bereits in Abb. fig:datasets gezeigt. Wir verzichten auf die kleinste Maßstabsebene, weil hier die segmentierten Regionen sehr klein sind. Bei einer Visualisierung mit der hier verwendeten Bildauflösung nehmen wir viele dieser Regionen nicht wahr.

Bei der visuellen Inspektion der Klassifikationsergebnisse von ADTboost in Abb. 4.19 ist erkennbar, wie sich die Regionen über mehrere Maßstäbe entwickeln. In der Referenz-Maßstabsebene $\sigma_m = 2$ (linke Spalte) sind vor allem relativ kleine Regionen enthalten. Die

Abbildung 4.19.: Klassifikationsergebnisse (in gelb) von ADTboost in vier verschiedenen Referenz-Maßstabsebenen σ_m bzgl. der Klassen `building` (oben), `window` (Mitte) und `vegetation` (unten).

durchschnittliche Größe der stabilen Regionen wächst bei größer werdendem Maßstab. Durch visuelle Inspektion kann man in den höchsten drei Maßstabsebenen nicht mehr die Regionen identifizieren, da viele benachbarte Regionen gleich klassifiziert wurden und daher die Grenzen der Regionen in dieser Präsentation nicht mehr sichtbar sind. Allerdings wissen wir aus der Größe der Datensätze, dass die Anzahl der stabilen Region in den beiden oberen Maßstabsebenen abnimmt. In Abb. 4.19 werden sowohl die Klassifikationserfolge als auch die Misserfolge sichtbar. Viele Gebäuderegionen der zweiten Maßstabsebene werden korrekt als `building` erkannt, viele Regionen aus den Büschen vor dem Haus als `vegetation`. In der höchsten Maßstabsebene werden die falschen Klassifikationsergebnisse durch die großen Flächen der Regionen besonders hervorgehoben. So wird beispielsweise die Himmelsregion als Gebäude klassifiziert. Ebenso wird eine Region als Vegetation erkannt, die wir in Abb. 3.8 als `mixture` visualisiert haben, da sich die Region über alle im Bild sichtbaren Pflanzen und die gesamte Straße sowie den schattigen Teil der Fassade direkt unterhalb des Dachs erstreckt.

Auf den anderen Datensätzen erzielen wir ähnlich zufrieden stellende Ergebnisse, die wir in Abb. 4.20 darstellen. Dabei zeigen wir einerseits die kleinen Dachregionen in unserem Beispiel-Luftbild aus dem Datensatz `aerial` und andererseits die kleinen Fassadenregionen und Fensterregionen für die Münchener Fassade des entzerrten Datensatzes `terra-3`, deren Fotos wir in Abb.2.14 gezeigt haben. Beim Luftbild in der obersten Reihe sind die vor allem die fehlerhaften Klassifizierungen in der dritten Maßstabsebene zu erwähnen, in Maßstabsebene vier werden zumindest die Regionen im Bereich des roten Dachs auch als solches erkannt. Bei den Fassadenregionen in der mittleren Reihe irrititiert uns vor allem die falsche Klassifikation der Himmelsregion in der obersten Maßstabsebene, bei den Fenstern in der untersten Reihe gibt es wieder überwiegend gute Klassifikationsergebnisse.

Wenn wir alle Regionen gleichen Klassifikationsergebnisses in einem Bild visualisieren, erhalten wir Klassifikationskarten, siehe Abb. 2.1. Diese geben wir in Abb. 4.21 nur für Objektteile und Instanzen selten vorkommender Klassen an. Bei Klassen, die große Teile des Bildes bedecken und zudem häufig als Klassifikationsergebnis ausgegeben werden, ist diese Darstellung nicht sinnvoll, da in diesen Fällen große Teile des Bildes entsprechend eingefärbt sind. In der oberen Reihe von Abb. 4.21 zeigen wir abwechselnd einen Luftbildausschnitt und anschließend die darin detektierten Regionen der Klasse `roof-tiles`, d.h. die kleinen Dachregionen. In der Reihe darunter zeigen wir die Detektionsergebnisse von vier verschiedenen Klassen (jeweils in anderen Bildern).

In Abb. 4.21 zeigen wir Klassifikationskarten mit zufriedenstellenden Ergebnissen. In allen Fällen sind Fehler erkennbar, d. h. nicht alle Instanzen der verwendeten Klassen wurden erfolgreich als solche erkannt. Zudem gibt es auch Verbesserungspotential bei der geometrischen Form der Regionen. In den Luftbildausschnitten fehlen häufig einzelne Dachflächen, bei der Fenster-Karte wurden viele Fenster nur unvollständig erkannt, d. h. es fehlen immer wieder einzelne Fensterscheiben.

In der dritten Präsentation visualisieren wir die klassifizierten Regionen in einem Bild zusammen, siehe Abb. 4.22. Dazu müssen wir den Alg. 3.1 lediglich bzgl. eines Punktes modifizieren: Anstelle der aus den Annotationen abgeleiteten Targets werden nun die Klassifikationsergebnisse zur Visualisierung der Semantik verwendet. Wir gehen wie gehabt vor und markieren erst die großen Regionen der obersten Maßstabsebene im Bild und danach die Regionen der kleineren Maßstabsebenen. Wir verwenden auch dieselbe Reihenfolge bei der Visualisierung der Klassen, um sicherzustellen, dass die komplexen Objekte wie Gebäude vorher im Ausgabebild visualisiert werden als deren Bestandteile, z. B. Fenster.

In Abb. 4.22 visualisieren wir die Ergebnisse aus vier verschiedenen Datensätzen, d. h. auch unter Verwendung verschiedener Klassen. Diese haben wir daher nur vertikal einheit-

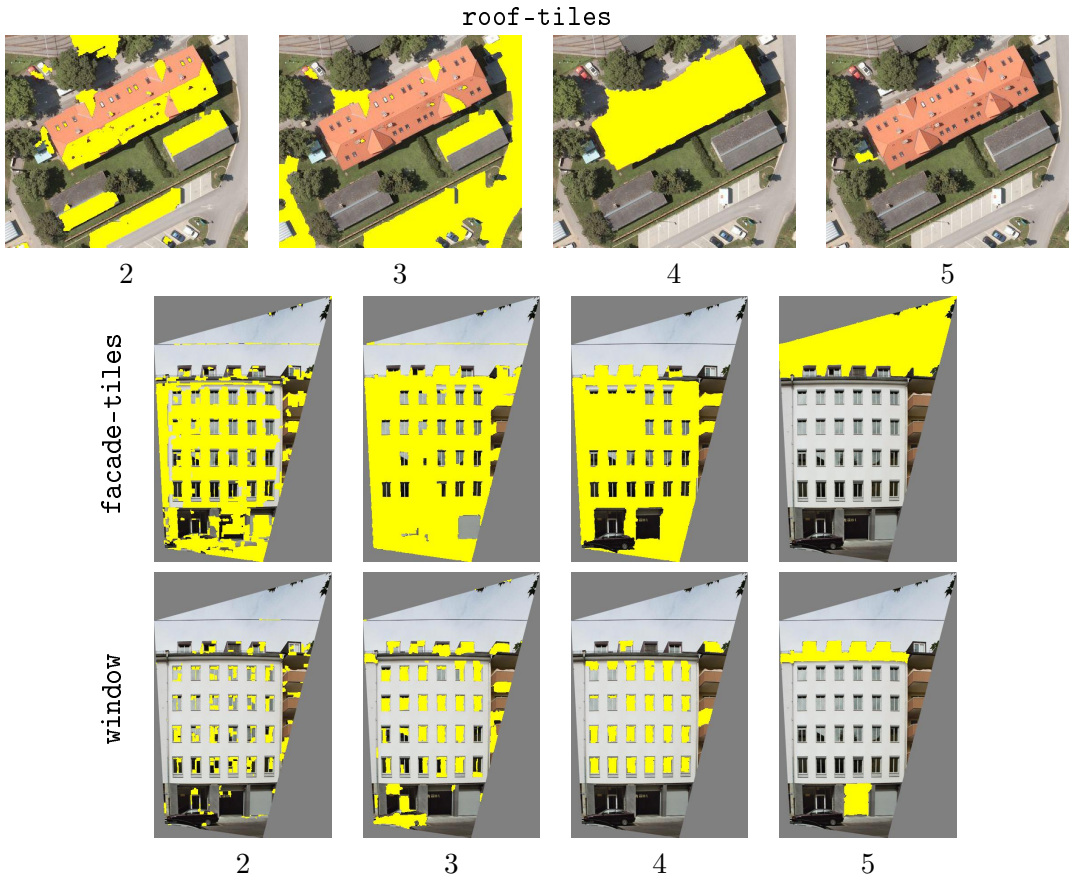


Abbildung 4.20.: Klassifikationsergebnisse (in gelb) von ADTboost in vier verschiedenen Referenz-Maßstabsebenen σ_m bzgl. der Klassen `roof-tiles` im Luftbild-Datensatz `aerial` (oben) sowie den Klassen `facade-tiles` (Mitte) und `window` (unten) im entzerrten Fassaden-Datensatz `terra-3`.

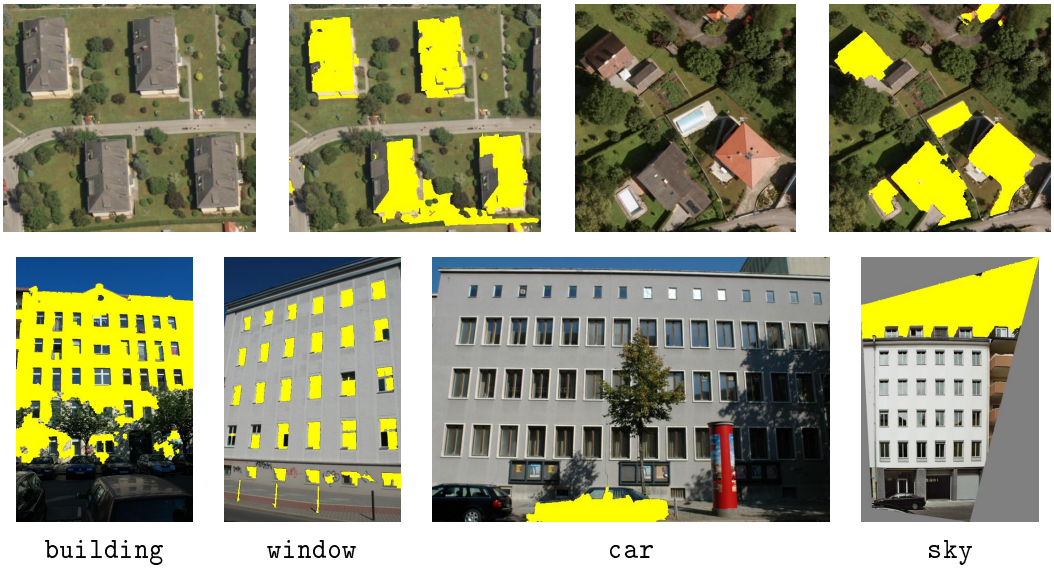


Abbildung 4.21.: ADTboost-Klassifikationskarten ausgewählter Klassen in Bildern verschiedener Datensätze.

lich verwendet, so dass man die Klassifikationsergebnisse untereinander und mit den Targets vergleichen kann. Weiße Pixel bedeuten dabei, dass in diesem Bildbereich keine stabilen Regionen segmentiert und daher auch nicht klassifiziert wurden. Schwarz sind alle Pixel, die als stabile Region in nicht annotierten Bereichen segmentiert wurden, d. h. sie gehören zur Klasse `background`. Pixel aus Regionen der für die Klassifikation zusätzlich eingeführte Klasse des Rests, `others`, stellen wir in den Bildern aller vier Datensätze gelb dar. Die anderen Farben unterscheiden sich spaltenweise in ihrer Semantik. In der linken Spalte zeigen wir das eine Beispielbild aus dem Benchmark-Datensatz `terra-1` mit rot für Gebäude, grün für Vegetation und blau für Fenster. In den beiden mittleren Spalten müssen diverse Klassen visualisiert werden. Hier ist vor allem vorzuheben, dass die hellgrauen Pixel zu Regionen der Klasse `facade` gehören, die Fenster sind blau markiert. In der rechten Spalte zeigen wir einen Luftbildausschnitt, in dem vor allem die vier Klassen `background` (schwarz), `others` (gelb), `roof` (rot), `mixture` (magenta) und `car` (grau) erkennbar sind.

Wir hatten bereits bei der Visualisierung der Targets auf die Schwierigkeit der Visualisierung von Regionen aus mehreren Maßstabsebenen hingewiesen, weil sich diese sehr stark überlappen. Die Klassifikationsergebnisse für die beiden Bilder aus den Datensätzen `terra-1` und `terra-3` in der ersten bzw. dritten Spalte von links in Abb. 4.22 bieten eine Darstellung, in denen wir viele übereinstimmende Farbzuweisungen erkennen. Das hatten wir nach Aufstellung der Statistik (ca. 50% richtig klassifiziert) anders erwartet. In der rechten Spalte wird die Dominanz der Klasse `background` sehr deutlich, da zu viele Regionen als Hintergrund erkannt wurden. Im Vergleich zwischen ADTboost und der Referenzklassifikation sieht man in beiden Zeilen viele korrekt klassifizierte Regionen und viele falsch klassifizierte. In den Bildern ist wie auch in der Statistik nicht erkennbar, dass beide Verfahren ähnlich gute bzw. schlechte Ergebnisse liefern.

Abbildung 4.22.: Visueller Vergleich zwischen den Targets der Regionen und den Klassifikationsergebnissen für die Beispielbilder aus Abb. 2.14, die wir noch einmal in der obersten Zeile zeigen. Darunter visualisieren wir erneut die Targets der Regionen (wie in Abb. 3.9). Die beiden unteren Zeilen sind neu, darin präsentieren wir die Klassifikationsergebnisse mit ADTboost sowie mit der MAP-Klassifikation im LDA-Unterraum. Vertikal gesehen bedeuten gleiche Farben auch gleiche Klassen. Weiße Pixel bedeuten dabei, dass in diesem Bildbereich keine stabilen Regionen segmentiert und daher auch nicht klassifiziert wurden. Schwarz sind alle Pixel, die als stabile Region in nicht annotierten Bereichen segmentiert wurden, d. h. sie gehören zur Klasse `background`. In den terrestrischen Aufnahmen wird Vegetation grün, die Fenster blau, die Restklasse gelb dargestellt, in der linken Spalte bedeutet rot Gebäude, in der Mitte grau Fassade. In den Luftbildern haben wir die Dächer rot visualisiert.

4.4.4. Zusammenfassung und Beurteilung

Wir haben in diesem Kapitel 65 regionenspezifische Merkmale definiert und die Klassifikationsmethode der Alternierenden Entscheidungsbäume auf den Mehrklassenfall verallgemeinert.

In unseren Experimenten haben wir dokumentiert, dass sich die klassenspezifischen Verteilungen vieler Merkmale unterscheiden und sich die Merkmale damit zur Klassifikation der Regionen eignen. Neben einer qualitativen Untersuchung der Merkmale haben wir auch Hellinger-Distanzen berechnet, um so die Unterscheidbarkeit der klassenspezifischen Verteilungen belegen zu können. Die Distanzen variieren zwischen den Klassen, erreichen aber Werte bis zu 0.67 (bei maximaler Hellinger-Distanz 1). In diesen Fällen können wir erwarten, dass die Klassen `car` und `window` sehr gut trennbar sind. Es gibt aber auch Klassenpaare mit niedrigen Hellinger-Distanzen, z. B. 0.15 für `building` und `vegetation`, d. h. diese Klassen sind schwieriger zu unterscheiden.

Für die Erweiterung der Alternierenden Entscheidungsbäume auf den Mehrklassenfall haben wir die binäre Variante analysiert und die entscheidenden Stellen für eine mögliche Erweiterung identifiziert. Um die Komplexität der Berechnungen für die Klassifikation niedrig zu halten, haben wir sehr einfache Klassifikator verwendet, die ADTboost durch eine hierarchische Anordnung zu einem besseren Klassifikator zusammensetzt. Bei diesen einfachen Klassifikatoren handelt es sich jeweils um einen Binärbaum mit mehreren Entscheidungen auf genau einem Merkmal. Somit können wir diese Klassifikatoren als achsenparallele Einteilungen des Merkmalsraums darstellen. Bei Vergleichen mit anderen Klassifikatoren auf Benchmark-Daten hat unser ADTboost-Verfahren zufriedenstellend abgeschnitten. Einige Verfahren haben Probleme mit den hochdimensionalen Merkmalsräumen bzw. mit den kleinen Datensätzen beim Trainieren des Klassifikators, ADTboost liefert diesbzgl. zuverlässig Ergebnisse. Allerdings schneidet er meistens schlechter ab als die Klassifikation mittels nächster Nachbarschaften oder mit dem ebenfalls linear trennenden Perceptron-Algorithmus. Dadurch sehen wir noch deutliches Verbesserungspotential für unseren Algorithmus.

Wir haben die Regionen mit zwei Methoden klassifiziert, mit einer MAP-Klassifikation im LDA-Unterraum sowie mit unserer Formulierung des Mehrklassen-ADTboost. Wir hatten in Kap. 2.3 sechs Kriterien für unsere Bildsegmentierung aufgeführt. Hier haben wir eben den Nachweis dafür gebracht, dass ADTboost das Potential hat, mehrere Klassen gleichzeitig zu unterscheiden. Dabei haben wir die Ausgabe des Klassifikators als Wahrscheinlichkeiten formuliert. Die einfachen Klassifikatoren, die wir bei ADTboost verwenden, nutzen diskrete, nicht weiter definierte Merkmalsverteilungen und können sowohl mit kleinen als auch mit großen Datenmengen operieren. Im folgenden Kapitel werden wir noch auf die letzte Anforderung, die Beurteilung der Merkmale durch den Klassifikator, eingehen. Bei unseren Experimenten auf den Gebäuderegionen haben wir die Daten bzgl. ihrer Maßstabszugehörigkeit getrennt klassifiziert. Beide Verfahren haben eine Erfolgrsate von etwas mehr als 50%, was uns nicht zufrieden stellt. Insbesondere bevorzugen beide Klassifikatoren die besonders häufig vorkommenden Klassen `building`, `window` und `vegetation`. Bei der Visualisierung der Klassifikationsergebnisse zeigt ADTboost allerdings viele zufriedenstellende Klassifikationsbilder.

5. Bedingtes Bayes-Netz für die hierarchische Bildinterpretation

In den beiden vorangegangenen Kapiteln haben wir unsere hierarchische Segmentierung eines Bildes und die Klassifikation der segmentierten Regionen durch Alternierende Entscheidungsbäume beschrieben. Bei der Klassifikation haben wir nur die regionenspezifischen Merkmale verwendet, bei denen wir den Kontext der Region relativ schwach berücksichtigt haben, z. B. bei den Differenzen zwischen den Farbwerten einer Region und ihrer Umgebung. Durch die Modellierung der Bildinterpretation mittels eines Bayes-Netzes wollen wir nun den Kontext aus der Hierarchie der Regionen in die Klassifikation der Regionen integrieren.

In diesem Kapitel werden wir die Struktur des Bayes-Netzes zusammenfassen. Anschließend erklären wir, wie wir die Übergangswahrscheinlichkeiten bestimmen. Und wir stellen den Inferenzalgorithmus vor, den Pearl (1997) für Bayes-Netze mit einfacher Baumstruktur formuliert hat. Zuletzt kombinieren wir unsere Ergebnisse aus der Klassifikation zu einer einheitlichen Bildinterpretation und präsentieren unsere Ergebnisse.

5.1. Struktur des Bedingten Bayes-Netzes

Die Komponenten des Bedingten Bayes-Netzes haben wir bereits in unserem Konzept in Kapitel 2.1 eingeführt. Es besteht aus M Knoten, die M Zufallsvariablen $\underline{x}_m$ unter der Bedingung $\boldsymbol{F}$ repräsentieren. Mit $\boldsymbol{F}$ beschreiben wir beobachtete Daten, z. B. die extrahierten Merkmale der segmentierten Regionen oder Informationen über den Bildmaßstab. Die Zufallsvariablen geben die Wahrscheinlichkeiten an, mit welcher die segmentierte Region S_m zur Klasse ω_k gehört. Demnach ist $\underline{x}_m$ eine diskrete Zufallsvariable mit K Zuständen. Die Struktur der M Knoten ist fast identisch mit der hierarchischen Struktur der M segmentierten Regionen, nur die Richtung der Kanten zwischen den Knoten wurde geändert. Dennoch übertragen wir die beiden in Kap. 2.1.2 eingeführten Funktionen π und χ für die Relationen zu den Eltern- bzw. Kinderknoten auf die bedingten Zufallsvariablen des Bedingten Bayes-Netzes.

Bei der hierarchischen Bildsegmentierung haben wir die Struktur bottom-up definiert, d. h. die Regionenhierarchie haben wir als Entwicklung der stabilen Regionen im Maßstabsraum definiert, ausgehend von der untersten Maßstabsebene und bis zur höchsten Maßstabsebene führend. Entsprechend hatten wir auch die Regionenhierarchie visualisiert: Mit Pfeilen von unten nach oben weisend. Im Bayes-Netz drehen wir diese Pfeile um: Wir formulieren die Abhängigkeiten zwischen den Knoten entsprechend einer Objekt-zu-Teil-Strategie. Wenn wir wissen, dass ein komplexes Objekt ein Gebäude ist, dann schließen wir auf seine Teile wie Fenster und Wandfläche. Analog formulieren wir die Abhängigkeiten zwischen den Zufallsvariablen: Wenn eine Region in einer großen Maßstabsebene wahrscheinlich ein Gebäude zeigt, so können wir auf die Teile dieser Region als Fenster oder Wandfläche schließen. In Abb. 5.1 zeigen wir ein Bedingtes Bayes-Netz, dessen Struktur zur Regionenhierarchie zu unserem Beispiel in Abb. 2.5 passt.

Unser Bedingtes Bayes-Netz besteht aus mehreren Bäumen, da wir i. d. R. mehrere Regionen in der höchsten Maßstabsebene segmentieren, die dann auch als stabile Regionen S_m

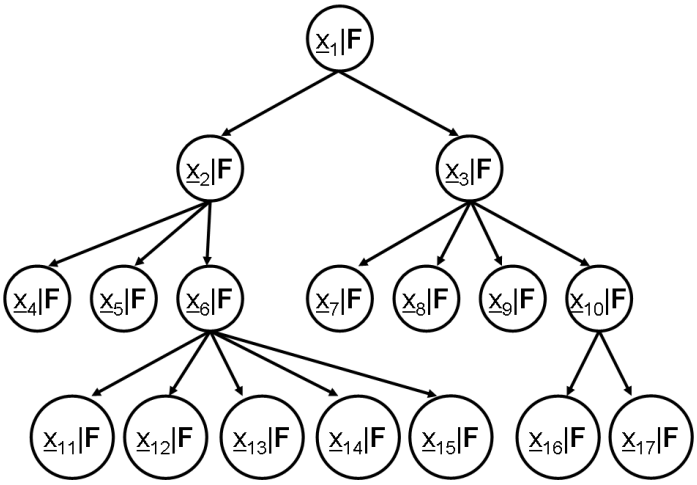


Abbildung 5.1.: Bedingtes Bayes-Netz für die Bildinterpretation.

gewertet werden. In jedem Baum können wir die Inferenz unabhängig von den Ereignissen in den anderen Bäumen propagieren. Dazu verwenden wir den Summe-Produkt-Algorithmus zur Inferenz in baumartigen Bayes-Netzen von Pearl (1997), der die folgenden vier Schritte abarbeitet:

1. die Aktualisierung der Überzeugungen einzelner Zufallsvariablen,
2. die Übertragung der Information von unten nach oben (bottom-up),
3. die Übertragung der Information von oben nach unten (top-down) und
4. Normalisierung zu Wahrscheinlichkeiten.

Der Algorithmus von Pearl (1997) setzt einen gerichteten und zyklenfreien Graphen voraus, was durch die baumförmige Struktur unseres Bayes-Netzes erfüllt wird. Durch die Summierungen bei der Bestimmung der Randverteilungen wird deutlich, dass die Merkmale diskret modelliert werden. Dies ist aber keine Beschränkung des Verfahrens, da man die Summen auch durch Integrale ersetzen kann, um mit kontinuierlichen Merkmalen arbeiten zu können. Es gibt auch keine Beschränkung auf univariate Merkmale, allerdings werden die Rechnungen aufwendiger, wenn statt univariaten Merkmalen hochdimensionale Merkmalsvektoren verarbeitet werden.

Wir geben den Summe-Produkt-Algorithmus entsprechend unserer Notation in Alg. 5.1 an. Ihm wird das Bayes-Netz mit seiner Struktur übergeben sowie die beiden Übergangswahrscheinlichkeiten. Die einen modellieren die Abhängigkeiten der Zufallsvariablen von den beobachteten Merkmalen $\boldsymbol{F}$ und sind daher unveränderlich. Wir verwenden sie zur Initialisierung der Zustände der Zufallsvariablen $\underline{x}_m$. Anschließend werden erst die Informationen von den Kindern zu den Eltern propagiert, siehe Zeile 13. Die Informationen der Kinder werden dabei multiplikativ verknüpft. Danach erfolgt die Propagierung der Information durch die Bäume von oben nach unten, siehe Zeile 19. Im Anschluss an jeden dieser beiden Teilschritte normalisieren wir die Zustandsbelegungen der Zufallsvariablen, um Wahrscheinlichkeiten abzuspeichern. Den Normalisierungsterm $\frac{1}{\zeta} = \frac{1}{P(\boldsymbol{F})}$ braucht man nur einmal zu berechnen, da er für alle Zufallsvariablen gleich ist, siehe (Bishop, 2006). Diese Wahrscheinlichkeiten geben wir zum Abschluss des Algorithmus aus.

Algorithm 5.1 Inferenz-Algorithmus für Bayes-Netze mit Baumstruktur

function BN-Inferenz($(\underline{x}_m)_1^M$, $(\boldsymbol{\beta}(m))_1^M$, $P(x_m \mid x_{\pi(m)})$)
  ▷ $\underline{x}_m$: M Zufallsvariablen in hierarchischer Anordnung
  ▷ $\boldsymbol{\beta}(m)$: M Klassifikationsergebnisse von ADTboost
  ▷ $P(x_m \mid x_{\pi(m)})$: Übergangswahrscheinlichkeiten
  ▷ Initialisierung der Zufallsvariablen mit ADTboost-Ergebnissen
  for $m = 1, \ldots, M$ **do**
    $P(x_m = \omega_k) = \beta_k(m)$
  end for
  ▷ Bottom-Up-Übertragung
  for $m = M, \ldots, 1$ **do**
    ▷ Starte bei den Blättern des Baumes
    ▷ C ist die Anzahl der Kinder der Zufallsvariable $\underline{x}_m$
    $P(x_m = \omega_k) = P(x_m = \omega_k) \cdot \prod_{c=1}^{C} \sum_{k'} P(x_{\chi_c(m)} = \omega_{k'}) P(x_{\chi_c(m)} = \omega_{k'} \mid x_m = \omega_k)$
  end for
  ▷ Top-Down-Übertragung
  for $m = 1, \ldots, M$ **do**
    ▷ Starte bei der Wurzel des Baumes
    if $\pi(m)$ existiert **then**
      $P(x_m = \omega_k) = P(x_m = \omega_k) \cdot \sum_{k'} P(x_m = \omega_k \mid x_{\pi(m)} = \omega_{k'}) P(x_{\pi(m)} = \omega_{k'})$
    end if
  end for
  Normalisierung der Wahrscheinlichkeiten der Zufallsvariablen
  return $\underline{x}_m$
end function

5.2. Übergangswahrscheinlichkeiten

In diesem Abschnitt gehen wir nun auf die Wahrscheinlichkeiten für das Auftreten von Klassenzugehörigkeiten hierarchisch benachbarter Regionen ein, die wir Übergangswahrscheinlichkeiten im Bedingten Bayes-Netz nennen und mit $P(\underline{x}_m \mid \underline{x}_{\pi(m)}, \boldsymbol{F})$ bezeichnen. Nach den Erfahrungen bei der Klassifikation der Regionen, diese Übergangswahrscheinlichkeiten ebenfalls in Abhängigkeit von dem Maßstabsbereich zu berechnen, in dem sich die segmentierte Region S_m befindet. Wir verwenden dabei wieder dieselben Maßstabsreferenzebenen σ_m wie bei der Klassifikation. Wir haben zwei verschiedene Übergangswahrscheinlichkeiten bestimmt. Die erste Art, $P(\underline{x}_m \mid \underline{x}_{\pi(m)}, \boldsymbol{F})$, verwendet bei $\boldsymbol{F}$ nur die Information bzgl. der Maßstabsreferenz σ_m. Die andere berücksichtigt zudem auch die Merkmalsausprägungen der beiden Regionen S_m und $S_{\pi(m)}$.

Bei einer Modellierung des Bedingten Bayes-Netzes mit Übergangswahrscheinlichkeiten ohne Berücksichtigung der Merkmale werden nur die Klassen $\widetilde{x}_m$ der segmentierten Regionen S_m berücksichtigt. Diese Übergangswahrscheinlichkeiten $P(\underline{x}_m \mid \underline{x}_{\pi(m)}, \boldsymbol{F})$ können wir durch Zählen der Relationen von $\widetilde{x}_m$ und $\widetilde{x}_{\pi(m)}$ und anschließendem Normieren unter Berücksichtigung von σ_m bestimmen. In diesem Fall sind die Übergangswahrscheinlichkeiten unabhängig von der Erscheinung der Regionen im Bild, d. h. die Wahrscheinlichkeiten für das Auftreten eines Fensters im Dach ist bei einer dunklen Region genauso hoch wie bei einer roten, wenn die Elternregion beider Regionen ein Dach ist. Das ist für uns nicht unbedingt plausibel, weshalb wir uns dafür entschieden haben, in einem zusätzlichen Experiment die Merkma-

le der Regionen bei der Bestimmung der Übergangswahrscheinlichkeiten zu berücksichtigen. Außerdem haben wir die schlechtere Performanz dieses Variante bzgl. des Klassifikationserfolges in (Drauschke & Förstner, 2011) dokumentiert. Wir gehen deshalb hier nur noch auf die Bestimmung der Übergangswahrscheinlichkeiten unter Verwendung der Merkmale ein.

5.2.1. Bayes-Netz mit Merkmalsintegration

Wir bezeichnen Übergangswahrscheinlichkeiten, bei denen wir die Merkmalsausprägungen der beiden Regionen S_m und $S_{\pi(m)}$ berücksichtigen, ebenfalls mit $P(\omega_m \mid \omega_{\pi(m)}, \boldsymbol{F})$, haben aber unter $\boldsymbol{F}$ drei Merkmale im Fokus: σ_m, f_m und $f_{\pi(m)}$. Dabei beschränken uns dabei aus Komplexitätsgründen auf genau ein (diskretisertes) Merkmal und genau eine Referenz-Maßstabsebene σ_m. Diese Merkmalsreduktion und -diskretisierung bildet keine Einschränkung der Voraussetzungen für die Modellierung von Bayes-Netzen sowie für die Durchführung des Summe-Produkt-Algorithmus von Pearl (1997), da dessen einzige Annahme ein zyklenfreier Graph mit gerichteten Kanten ist. Allerdings führen sowohl die Merkmalsreduktion als auch die Merkmalsdiskretisierung zu einem Informationsverlust, d. h. die Übergangswahrscheinlichkeiten könnten besser mit kontinuierlichen und hochdimensionalen Merkmalen modelliert werden. Das allerdings führt zu komplexeren Implementierungen, und es muss beachtet werden, dass die Übergangswahrscheinlichkeiten aus genügend großen Stichproben abgeleitet werden. Bei der Analyse der extrahierten Merkmale haben wir festgestellt, dass wir diese durch verschiedene Verteilungsfunktionen modellieren müssen, siehe Kapitel 4.1. Daher haben wir bei der Implementierung der Alternierenden Entscheidungsbäume auf die Definition der Merkmale mittels kontinuierlicher Verteilungen verzichtet. Ebenso greifen wir auch nun auf diskrete Merkmale zurück, die wir aus den Histogrammen ableiten.

Eine Einteilung des Merkmalsraums in gleichgroße Intervalle ist bei den meisten Merkmalen ungünstig, da zahlreiche Merkmale unterschiedlich stark besetzte Histogramme aufweisen, insbesondere die normal- und exponentialverteilten Merkmale. Wir diskretisieren daher die Merkmalsverteilungen so, dass wir die Regionen dadurch nahezu gleich auf die Intervalle aufteilen, d. h. wir bestimmen die $k-\alpha$-Quantile der Verteilungen. Praktisch projizieren wir dazu die Daten in viele gleichbreite, aber kleine Intervalle zwischen der minimalen und der maximalen Ausprägung des Merkmals und bestimmen das kumulative Histogramm, woraus wir effizient $k-\alpha$-Quantile eines Merkmals ableiten. In Abb. 5.2 zeigen wir diese Vorgehensweise für eine Einteilung in fünf α-Quantile bzw. Intervalle der Merkmalsausprägungen.

Die Übergangswahrscheinlichkeiten bestimmen wir ausgehend von einer Region S_m in Abhängigkeit von ihrer Maßstabsreferenzebene σ_m und den beiden Merkmalsausprägungen der Regionen S_m und $S_{\pi(m)}$, die jeweils einen von fünf Werten annehmen. Da wir die Maßstabsreferenzebene analog zur Klassifikation, siehe Kap. 4.2.1, bestimmen, unterscheiden wir zwischen fünf verschiedenen Maßstabsebenen. Somit unterscheiden wir $5^3 = 125$ verschiedene Fälle, um die Übergangswahrscheinlichkeiten zu bestimmen. Die Übergangswahrscheinlichkeiten erhalten wir durch Zählen der entsprechenden Vorkommen der beiden Klassen $\widetilde{x}_m$ und $\widetilde{x}_{\pi m}$, so dass wir die 125 verschiedenen Tabellen mit den Wahrscheinlichkeiten $P(\widetilde{x}_m, \widetilde{x}_{\pi(m)} \mid \boldsymbol{F})$ aufstellen könnten. Stattdessen normieren wir gleich die Werte in den Tabellen derart, dass wir die Wahrscheinlichkeiten $P(\widetilde{x}_m \mid \widetilde{x}_{\pi(m)}, \boldsymbol{F})$ enthalten.

In Tab. 5.1 zeigen wir die gemittelten Übergangswahrscheinlichkeiten für die Maßstabsreferenzebene $\sigma = 3$. Gemittelt haben wir hierbei über die fünf Tests der Kreuzvalidierung und über die 5×5 verschiedenen Möglichkeiten für die Merkmalsausprägungen der beiden Regionen. In der Hauptdiagonale haben wir die Tabelleneinträge fett markiert. Dadurch wird schnell erkennbar, dass die Region S_m mit hoher Wahrscheinlichkeit zur Klasse `car` bzw.

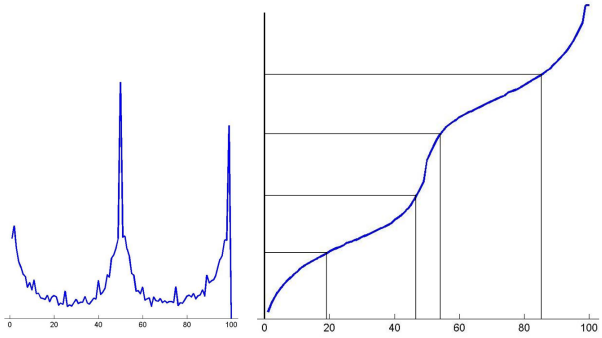


Abbildung 5.2.: Einteilung des Merkmalsraums in fünf Intervalle: Histogramm eines Merkmals (links) und dessen kumulatives Histogramm (rechts). Im kumulativen Histogramm kann man die Intervallgrenzen effizient bestimmen, so dass die Daten sich zu gleichen Teilen auf die Intervalle verteilen.

Tabelle 5.1.: Gemittelte Übergangswahrscheinlichkeiten der dritten Maßstabsreferenzebene. Spaltenweise geben wir die Klasse der Elternregion $S_{\pi(m)}$ angegeben, zeilenweise die der Region S_m. Die Einträge 0 zeigen an, dass diese Klassenpaarung nicht vorkommt, die Einträge 0.000 sind das Ergebnis einer Rundung.

Klasse	`others`	`build.`	`build.-mix`	`car`	`veget.`	`window`	`backg.`
`others`	**0.340**	0.077	0.013	0.006	0.056	0.006	0.007
`building`	0.185	**0.358**	0.091	0.010	0.022	0.021	0.000
`building-mix`	0.067	0.221	**0.721**	0.039	0.009	0.200	0
`car`	0.030	0.071	0.018	**0.917**	0.004	0.026	0.002
`vegetation`	0.095	0.075	0.015	0.016	**0.781**	0.004	0.006
`window`	0.138	0.148	0.143	0.011	0.009	**0.743**	0
`background`	0.145	0.050	0	0.015	0.118	0.000	**0.983**

`vegetation` gehört, wenn auch die Elternregion dieses Klassenlabel trägt.

5.2.2. Merkmalsselektion

Im vorherigen Abschnitt haben wir beschrieben, wie die Übergangswahrscheinlichkeiten unter Verwendung von beobachteten Merkmalen f_m und $f_{\pi(m)}$ berechnet werden. Wir benutzen dabei nur ein Merkmal, um die Auswertung im Bayes-Netz überschaubar zu behalten. Dies erfordert aber eine Merkmalsauswahl. In (Drauschke & Förstner, 2011) haben wir durch eine vollständige Suche das beste Merkmal bestimmt, das auf den Trainingsdaten zu den die besten Verbesserungen bei den Klassifikationsergebnissen führte. Dieses Vorgehen haben wir hier durch eine Merkmalsselektion ersetzt.

Nach Mladenić (2006) können die Methoden der Merkmalsauswahl in zwei wesentliche Kategorien eingeteilt werden: in Filter und in Wrapper. Filter-Methoden analysieren die Relevanz von Merkmalen unabhängig von der Klassifikationsmethode, z. B. durch Ranking-Verfahren wie Korrelationskoeffizienten (Guyon & Elisseeff, 2003).

Wrapper-Methoden dagegen evaluieren die Merkmale hinsichtlich der Klassifikationsergebnisse. Drauschke & Förstner (2008) haben eine Bewertung von Merkmalen vorgeschlagen, die

Tabelle 5.2.: Ergebnisse der Merkmalsselektion mit ADTboost. Wir zeigen die zehn besten Merkmale mit ihrem Einfluss r auf die Klassifikation mit ADTboost, nach den verwendeten Maßstabsreferenzebenen unterschieden.

Rang	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$	$\sigma = 5$
1	$f_{10} : 9.49$	$f_{10} : 8.91$	$f_{10} : 6.59$	$f_{10} : 7.21$	$f_{10} : 11.40$
2	$f_{15} : 6.60$	$f_{15} : 5.62$	$f_{32} : 6.26$	$f_{41} : 6.77$	$f_{8} : 7.48$
3	$f_{14} : 4.22$	$f_{32} : 4.50$	$f_{28} : 3.89$	$f_{32} : 6.57$	$f_{5} : 7.11$
4	$f_{11} : 3.11$	$f_{36} : 3.76$	$f_{41} : 3.77$	$f_{8} : 5.24$	$f_{28} : 6.17$
5	$f_{53} : 2.94$	$f_{14} : 3.27$	$f_{52} : 3.35$	$f_{28} : 4.47$	$f_{32} : 6.00$
6	$f_{60} : 2.77$	$f_{52} : 3.09$	$f_{15} : 3.28$	$f_{5} : 3.57$	$f_{7} : 5.46$
7	$f_{36} : 2.75$	$f_{41} : 2.89$	$f_{36} : 2.87$	$f_{36} : 3.38$	$f_{36} : 5.45$
8	$f_{41} : 2.73$	$f_{50} : 2.69$	$f_{14} : 2.39$	$f_{14} : 2.75$	$f_{46} : 5.40$
9	$f_{50} : 2.64$	$f_{25} : 2.52$	$f_{18} : 2.34$	$f_{18} : 2.58$	$f_{40} : 4.31$
10	$f_{22} : 2.60$	$f_{35} : 2.15$	$f_{17} : 2.07$	$f_{39} : 2.47$	$f_{41} : 4.16$

auf dem Einfluss r der Merkmale auf das Klassifikationsergebnis von ADTboost beruht. Das Prinzip funktioniert wie folgt: Wir haben unsere einfachen ADTboost-Klassifikatoren κ_t so definiert, dass sie jeweils genau ein Merkmal f_d verwenden. Den Klassifikationserfolg eines Klassifikators κ_t, gemessen an seinen B Gewichten $\boldsymbol{\alpha}_t{}^b$, können wir demnach auch dem Merkmal f_d zuordnen. Kumuliert man nun alle die Erfolge aller einfachen Klassifikatoren, d. h. wir addieren diese Gewichte $\boldsymbol{\alpha}_t{}^b$ für alle T Klassifikatoren κ_t und über alle B Bereiche $\mathfrak{b}_b$, dann erhalten wir den Einfluss r eines Merkmals auf die Gesamtklassifikation:

$$r(d) = \sum_{t=1}^{T} \sum_{b=1}^{B} \mathfrak{p} \boldsymbol{\alpha}_t{}^b \mathbb{I}(\kappa_t \text{ verwendet Merkmal } f_d). \tag{5.1}$$

Dabei berücksichtigen wir auch, dass die Klassifikatoren hierarchisch angeordnet werden: κ_t verwendet Merkmal f_d schließt die Vorbedingung und die eigentliche Klassifikation mit ein. Der Faktor $\mathfrak{p}$ gibt den Anteil der Daten an, die beim Lernen der Klassifikation berücksichtigt wurden. So werden die vergleichsweise hohen Gewichte in den Blättern des Alternierenden Entscheidungsbaums wieder klein skaliert. Gibt ADTboost einen Baum zurück, in dem nur ein Klassifikator an den Wurzelknoten gehängt wurde, dann ist das Merkmal dieses Klassifikators das Beste, weil es von allen einfachen Klassifikatoren verwendet wird: von κ_1 direkt und von allen anderen einfachen Klassifikatoren als Vorbedingung.

Wir haben die Klassifikation mittels einer Kreuzvalidierung analysiert, so dass jedes Bild genau einmal zu den Testdaten gehört. Für ein gegebenes Testbild ist aber das Klassifikationsmodell und damit auch das am besten geeignete Merkmal nach Gl. 5.1 eindeutig bestimmbar. Um Auskunft über die Merkmale im Allgemeinen zu erhalten, haben wir daher die Bewertung des Einflusses eines Merkmals durch Mittelung der Rangbewertung $r(d)$. Die Ergebnisse der besten zehn Merkmale je Maßstabsreferenzebene geben wir in Tab. 5.2 an.

Bei unseren Experimenten mit Datensatz `terra-1` haben wir festgestellt, dass bei allen Tests und in allen Skalen das beste Merkmal die relative Höhenposition des Mittelpunkts der Bounding Box im Bild (Index 10) ist. Erst bei den nächstbesten Merkmalen gibt es Veränderungen bei der Art des Merkmals. In den beiden untersten Skalen sind die beiden Farbmerkmale mittlerer Blauwert (Index 14) und mittlerer Huewert (Index 15) sehr gut. Bei den beiden nächsthöheren Skalen spielen die Entropie des Gradientenhistogramms im grünen Kanal (Index 32) und der größere Eigenwert der Momentenmatrix des generalisierten Vierecks

(Index 41)eine wichtige Rolle. Und in der obersten Skala haben die Breite der Bounding Box (Index 8) und der Umfang einer Region (Index 5) hohen Einfluss auf die Klassifikation. Diese Ergebnisse sind sehr gut nachvollziehbar. In den kleinen Skalen sind viele Objekte noch extrem übersegmentiert, d. h. die Regionen zeigen meist nur einen Ausschnitt des Objekts. Entsprechend ist die Farbe einer Region besonders aussagekräftig. In den mittleren Skalen sind die Regionen schon größer, so dass die Gradienteninformation gute Erkenntnisse über die Regionen liefern. In der obersten Skala wiederum stellen die Regionen oft komplexe Objekte dar, so dass sich die Klassifikation auf dem Umfang und anderen Formmerkmalen stützt. Das beste Texturmerkmal befindet sich in den mittleren Skalen erst an viertbester Stelle und hat einen ähnlichen Evaluationswert $r(d)$ wie die ähnlich platzierten Merkmale, d. h. das beste Texturmerkmal ist ähnlich gut wie das beste Merkmal bzgl. der Gradientenhistogramme.

Während sich die besten Merkmale bei dem etwas größeren Datensatz `terra-2` von den besten Merkmalen des Benchmark-Datensatzes `terra-1` nicht unterscheidenden, gibt es einige Veränderungen bei der Aufstellung der besten Merkmale bzgl. der Regionen aus den entzerrten Bildern des Datensatzes `terra-3`. Auf den drei unteren Skalen ist nach wie vor die relative Höhenposition, d. h. das Merkmal 10 das Beste. Aber in den beiden oberen Regionen sind Merkmal 41 bzw. Merkmal 39 (der Eintrag der Momentenmatrix auf der Nebendiagonale) die besten Merkmale. Das liegt vor allem daran, dass viele rechteckig geformte Objekte in den Bildern gezeigt werden und nun Regionen ebenso geformt sind. Ansonsten spielen auch hier wieder Farbmerkmale in den unteren Skalen, Gradientenmerkmale in den mittleren Skalen und Formmerkmale in den oberen Skalen eine besondere Rolle.

Bei den Luftbildausschnitten des Datensatzes `aerial` sind teilweise ganz andere Merkmale die besten. Beim Blick von Oben korreliert die Höhenposition der Bounding Box einer Region nicht mehr mit den Klassen, so dass Farbmerkmale (mittler Huewert als Merkmal 15), Merkmale des Gradientenhistogramms (Entropie im roten und grünen Kanal als Merkmale 28 und 32) und abgeleitete Merkmale aus dem generalisierten Viereck (der Eintrag der Momentenmatrix auf der Nebendiagonale als Merkmal 39) die besten sind. Wir finden nur ein Basis- bzw. Formmerkmal unter den besten fünf Merkmalen, und das ist die Fläche einer Region bei der Klassifikation in der zweitobersten Skala.

5.3. Experimente

Wir haben das Bedingte Bayes-Netz auf unseren vier Datensätzen getestet und dokumentieren nun die Ergebnisse dieser Experimente. Das gliedern wir in drei Teile. Zuerst geben wir Auskunft über die statistischen Angaben, d. h. die Konfusionsmatrizen im Vergleich zur Klassifikation ohne Integration von Kontext durch die Bestandteilshierarchie (ADTboost). Dann visualisieren wir die klassifizierten Regionen analog zu Alg. 3.1. Zum Schluss vergleichen wir die Farbgebung der Pixel dieser Klassifikationsbilder und präsentieren diese Ergebnisse.

5.3.1. Statistische Auswertung

Wir stellen die Konfusionsmatrizen mit den relativen Angaben für den Klassifikationserfolg je Klasse im Anhang A.3 zusammen, da wir dabei die Ergebnisse aller vier Datensätze präsentieren. Hier fassen wir die Ergebnisse für den Benchmark-Datensatz `terra-1` zusammen und gehen auf die Veränderung bzgl. der alleinigen ADTboost-Klassifikation ein.

Generell verbessert sich der Klassifikationserfolg von 54.9% auf 60.7% bei sieben Klassen. Diese Verbesserung entsteht allerdings durch eine weitere Bevorzugung der Klasse `building`, deren klassenspezifischer Klassifikationserfolg über 83% liegt. Dagegen kommen die Klassen

Tabelle 5.3.: Differenzen der Klassifikationsraten bzgl. des Datensatzes `terra-1`. Wir geben die Klassifikationserfolge des Bedingten Bayes-Netzes im Vergleich mit den ADTboost-Ergebnissen an.

Klasse	others	build.	build.-mix	car	veget.	window	backg.
others	$-\mathbf{3.9}\%$	$+17.4\%$	-0.3%	-1.8%	-4.6%	-5.9%	-1.0%
building	-1.7%	$+\mathbf{14.9}\%$	-0.5%	-0.2%	-4.5%	-7.7%	-0.3%
building-mix	-1.1%	$+7.2\%$	$-\mathbf{1.0}\%$	-0.2%	-2.9%	-1.7%	-0.2%
car	-2.9%	$+32.1\%$	-0.6%	$-\mathbf{5.2}\%$	-7.9%	-13.4%	-2.2%
vegetation	-1.1%	-0.6%	-0.2%	-0.3%	$+\mathbf{5.9}\%$	-3.2%	-0.6%
window	-0.6%	$+4.3\%$	-0.6%	-0.2%	-3.7%	$+\mathbf{0.9}\%$	-0.1%
background	-0.5%	$+8.3\%$	-0.1%	-1.2%	$+1.9\%$	-6.3%	$-\mathbf{2.1}\%$

`car`, `building-mixture` und `background` als Klassifikationsresultat praktisch nicht mehr vor: 46 der 131 060 Regionen wird das Klassenlabel von einer der drei Klassen zugeordnet, in 17 Fällen stimmt auch das Ergebnis.

In Tab. 5.3 zeigen wir die relativen Differenzen zur ADTboost-Klassifikation. Jede Zeile zeigt die tatsächliche Klasse $\widetilde{x}_m$ der Regionen an, jede Spalte das Klassifikationsergebnis $\widehat{x}_m$. Hier ist der deutliche Anstieg bei der Zuweisung zur Klasse `building` zu erkennen. Die Klassifikationserfolge für die beiden ebenfalls sehr häufig vorkommenden Klassen `vegetation` und `window` fallen ebenfalls besser aus, wenn auch nur leicht.

Bei den anderen drei Datensätzen wiederholt sich dieses Bild. Beim Datensatz `terra-2` steigt der Erfolg der Klassifikation mit acht Klassen von 33.6% auf 39.0%, beim Datensatz `terra-3` von 42.0% auf 46.6% bzgl. der Klassifikation mit acht Klassen, und auf dem Luftbilddatensatz `aerial` bleibt der Klassifikationserfolg nahezu unverändert, steigt von 51.4% auf 51.5% bei Verwendung von sechs Klassen. In allen drei Fällen entsteht die Verbesserung des Klassifikationsergebnisses zu Ungunsten der selten vorkommenden Klassen.

5.3.2. Visualisierung der Ergebnisse

Diese Bevorzugung der häufig vorkommenden Bilder schlägt sich auch bei der Visualisierung der Bildinterpretation nieder. In den Bildern zu Datensatz `terra-1` gibt es häufig nur noch drei Farben für die Visualisierung von Gebäuden, Fenstern und Vegetation. Den modifizierten Alg. 3.1 für die Darstellung der Klassifikationsergebnisse mit ADTboost können wir an dieser Stelle wieder verwenden. Entsprechend markieren wir erst die großen Regionen der obersten Maßstabsebene im Bild und danach die Regionen der kleineren Maßstabsebenen. Ebenso verwenden wir dieselbe Reihenfolge bei der Visualisierung der Klassen, um sicherzustellen, dass die komplexen Objekte wie Gebäude vorher im Ausgabebild visualisiert werden als deren Bestandteile, z. B. Fenster.

In Abb. 5.3 visualisieren wir die Ergebnisse aus den vier verschiedenen Datensätzen, d. h. auch unter Verwendung verschiedener Klassen. Diese haben wir daher nur in vertikaler Anordnung einheitlich verwendet, so dass man die Klassifikationsergebnisse von ADTboost bzw. durch das Bedingte Bayes-Netz untereinander und mit den Klassifikationszielen, den Targets, vergleichen kann. Die Target-Visualisierung zeigt dabei an, wie das Bild aussehen müsste, wenn die Klassifikation perfekt wäre.

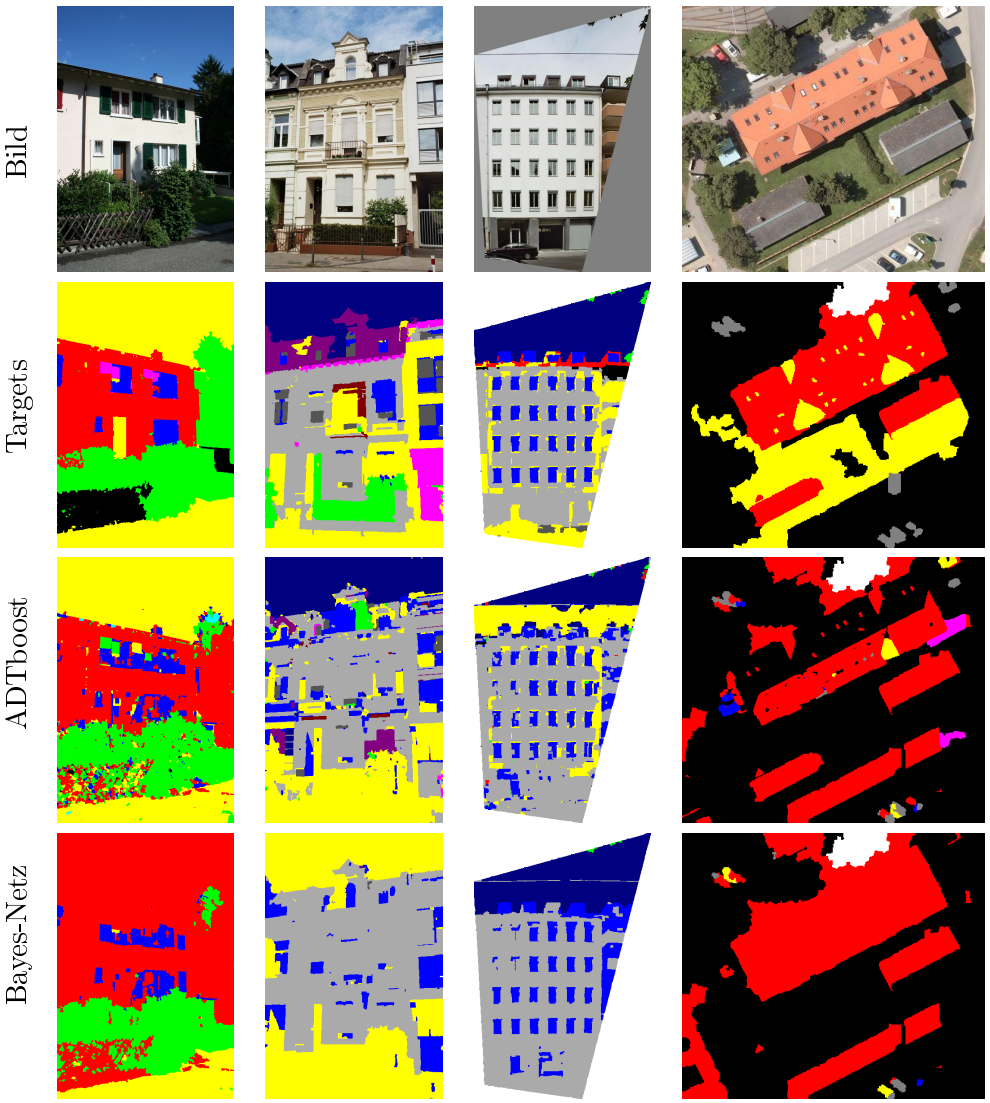


Abbildung 5.3.: Visueller Vergleich zwischen den Targets der Regionen und den Klassifikationsergebnissen mit ADTboost bzw. mit dem Bedingten Bayes-Netz für die Beispielbilder aus Abb. 2.14, die wir noch einmal in der obersten Zeile zeigen. Darunter visualisieren wir erneut die Targets der Regionen (wie in Abb. 3.9) und die ADTboost-Ergebnisse (wie in Abb. 4.22). In der untersten Zeile präsentieren wir die Klassifikationsergebnisse mit dem Bedingten Bayes-Netz. Vertikal gesehen bedeuten gleiche Farben auch gleiche Klassen. Weiße Pixel bedeuten dabei, dass in diesem Bildbereich keine stabilen Regionen segmentiert und daher auch nicht klassifiziert wurden. Schwarz sind alle Pixel, die als stabile Region in nicht annotierten Bereichen segmentiert wurden, d. h. sie gehören zur Klasse `background`. In den terrestrischen Aufnahmen wird Vegetation grün, die Fenster blau, die Restklasse gelb dargestellt, in der linken Spalte bedeutet rot Gebäude, in der Mitte grau Fassade. In den Luftbildern haben wir die Dächer rot visualisiert.

Tabelle 5.4.: Relative Übereinstimmung der Bildinterpretationen im pixelweisen Vergleich

Vergleich	terra-1	terra-2	terra-3	aerial
Annot. vs. Targets:	48.8%	66.9%	78.7%	89.0%
Annot. vs. ADTboost:	28.8%	48.4%	61.2%	82.1%
Annot. vs. BayesNetz:	27.7%	51.1%	68.7%	84.5%
Targets vs. ADTboost:	68.4%	48.4%	58.4%	80.2%
Targets vs. Bayes-Netz	70.1%	53.8%	65.0%	81.7%

5.3.3. Pixelweise Evaluation

Durch die verschiedenen Visualisierungen der Bildinterpretationen wie die manuellen Annotationen, die Targets der Regionen und die Klassifikationsergebnisse von ADTboost und durch das Bedingte Bayes-Netz können wir die Ergebnisse auch pixelweise evaluieren. Insgesamt haben wir die folgenden fünf Vergleiche durchgeführt:

1. Annotationen mit den Klassifikationszielen (Targets)
2. Annotationen mit ADTboost-Ergebnissen
3. Annotationen mit Ergebnissen des Bedingten Bayes-Netzes
4. Targets mit mit ADTboost-Ergebnissen
5. Targets mit Ergebnissen des Bedingten Bayes-Netzes.

Die durchschnittliche Übereinstimmung bzgl. aller vier Datensätze geben wir in Tab. 5.4 an. Die wichtigste Zeile ist hierbei die dritte, die den Vergleich zwischen den Annotationen und den Klassifikationsergebnissen durch das Bayes-Netz angibt. Wir führen aber eine überwachte Klassifikation durch, die den Klassifikator zur Erkennung der Klassenlabel der Targets trainiert. Daher ist zur Bewertung des Bedingten Bayes-Netzes ein Vergleich zwischen den Targets und den Klassifikationsergebnissen des Bayes-Netzes besser. Für einen Vergleich mit den Annotationen müssten wir die Klassifikationsziele unabhängig von den Targets und dem Algorithmus zur Visualisierung der Bildinterpretation betrachten, was ausgeschlossen ist.

In Tab. 5.4 erkennen wir, dass der Benchmark-Datensatz am schlechtesten abschneidet, wenn wir die automatisch generierten Bildinterpretationen mit den manuell erstellten Annotationen vergleichen. Deutlich besser sind bei diesem Datensatz dagegen die Vergleiche zwischen den Klassifikationsergebnissen und den Targets. Die besten Bildinterpretationen erzielen wir anscheinend auf den Luftbildausschnitten. Allerdings muss hier berücksichtigt werden, dass in vielen Bildern mehr als 75% der Pixel nicht annotiert wurden, weil sie Vegetation oder Straße darstellen. Hierbei wirkt sich die Bevorzugung der sehr häufig vorkommenden Klasse `background` positiv auf die Beurteilung aus.

Fernerhin haben wir für jedes Bild der verwendeten Datensätze die Übereinstimmungen im pixelweisen Vergleich bestimmt. In den Abb. 5.4, 5.5 und 5.6 zeigen wir Bilder, bei denen der pixelweise Vergleich hohe Übereinstimmungen zwischen den Farbgebungen der Targets und den Farbgebungen der Klassifikationsergebnisse durch das Bedingte Bayes-Netz anzeigt. Die Semantik der Farben ist dabei dieselbe wie in den anderen Abbildungen, z. B. Abb. 5.3. Innerhalb einer Abbildung haben wir immer Bilder des gleichen Datensatzes verwendet, somit bedeuten die Farben innerhalb jeder dieser Abbildungen auch dasselbe. Beispielsweise zeigt in Abb. 5.4 rot Gebäude an, Fenster sind blau und Vegetation grün.

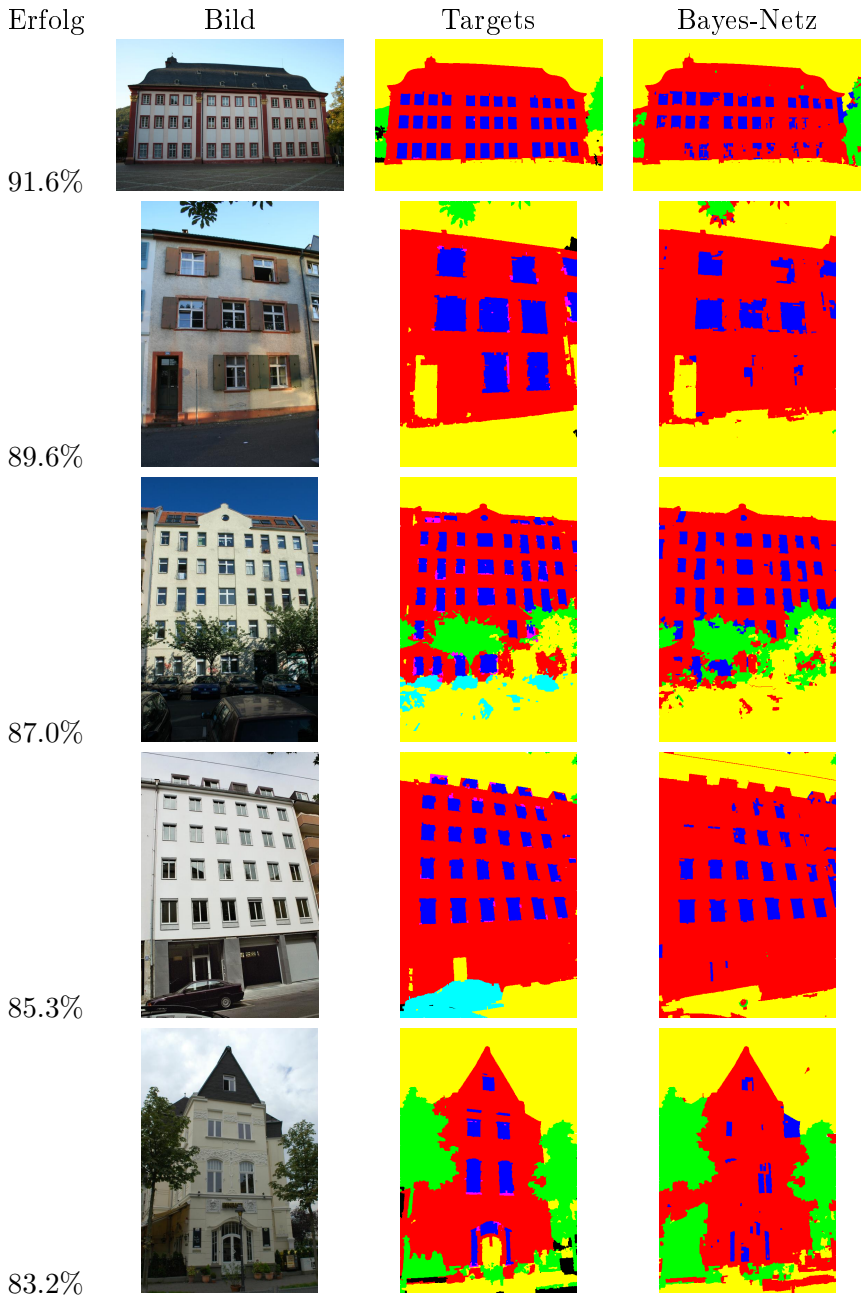


Abbildung 5.4.: Ausgewählte Ergebnisse im pixelweisen Vergleich für Datensatz `terra-1`. Die Farben geben die Semantik der Szene wider: Gebäude (rot), Fenster (blau), Vegetation (grün). Gelb stellt Regionen der Restklasse dar, die u. a. Himmel und Boden vereint.

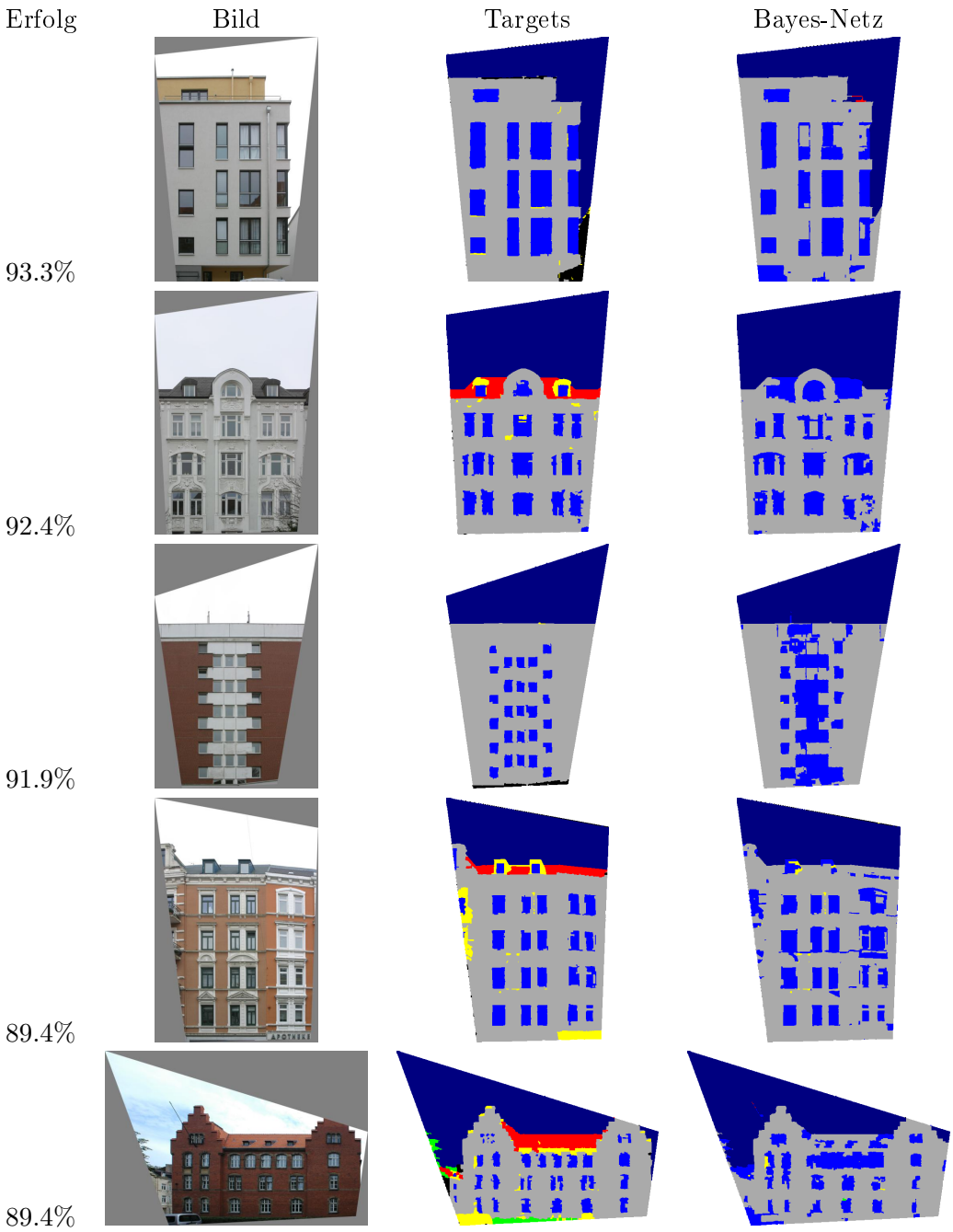


Abbildung 5.5.: Ausgewählte Ergebnisse im pixelweisen Vergleich für Datensatz `terra-3`. Die Farben geben die Semantik der Szene wider: Fassaden (grau), Fenster (blau), Himmel (dunkelblau), Dach (rot). Gelb stellt Regionen der Restklasse dar.

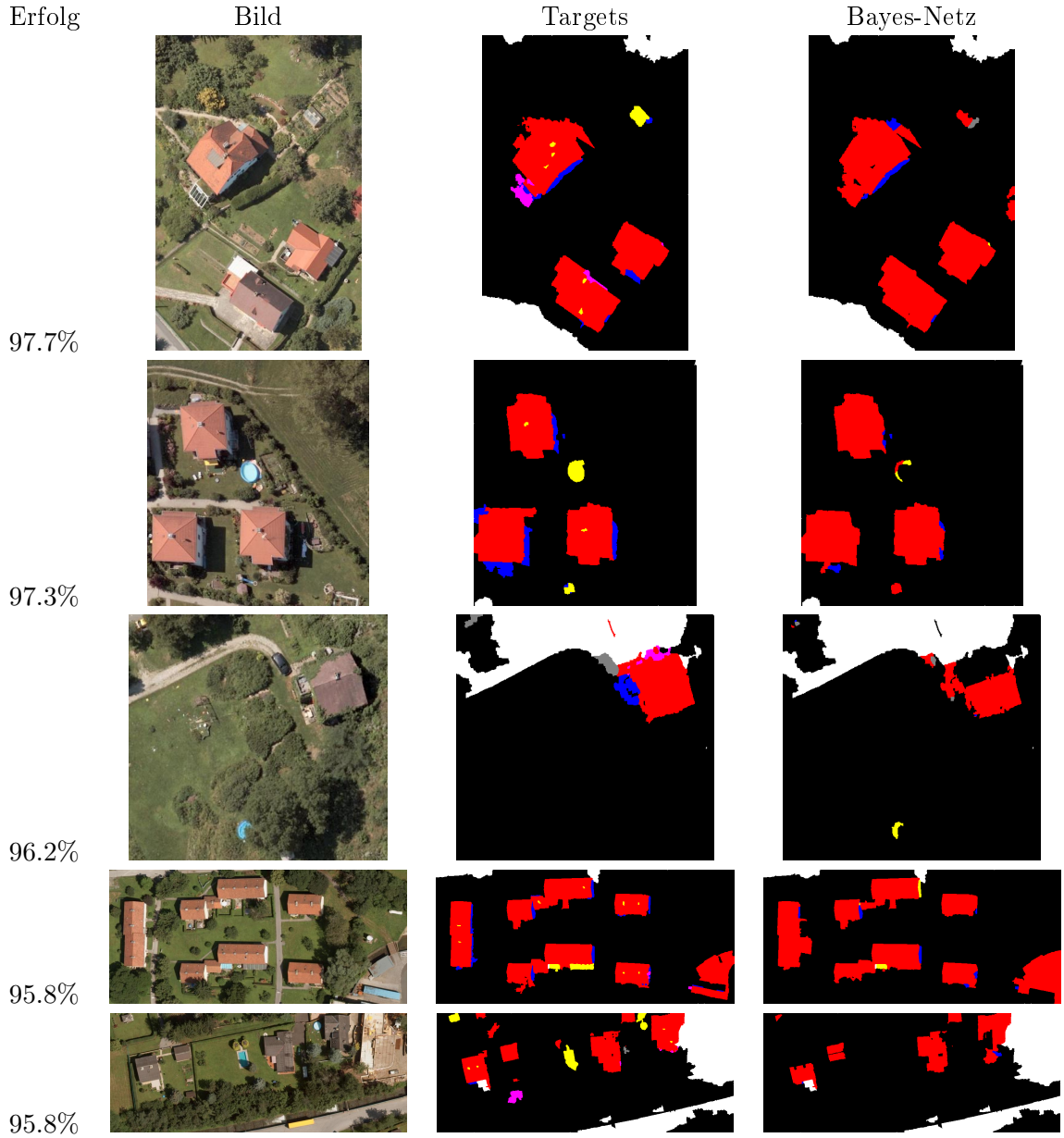


Abbildung 5.6.: Ausgewählte Ergebnisse im pixelweisen Vergleich für Datensatz `aerial`. Die Farben geben die Semantik der Szene wider: Dach (rot), Gebäude (blau), Mixturen (magenta). Gelb stellt Regionen der Restklasse dar.

5.3.4. Zusammenfassung und Beurteilung

Wir haben in diesem Kapitel unser Bedingtes Bayes-Netz konstruiert, dessen Struktur wir aus der hierarchischen Segmentierung ableiten und in das wir die Klassifikationsergebnisse von ADTboost integrieren. Die Übergangswahrscheinlichkeiten haben wir in Abhängigkeit von einer Maßstabsebene und von beobachteten Merkmalen durch Auswertung der Klassenzugehörigkeiten hierarchisch benachbarter Regionen bestimmt. Für die Inferenz im Bayes-Netz haben wir den Algorithmus von Pearl (1997) angewendet, da unser Bayes-Netz eine baumförmige Struktur aufweist.

In unseren Experimenten haben wir festgestellt, dass sich zwar der Klassifikationserfolg durch das Bedingte Bayes-Netz verbessert, dieser aber durch eine erneute Bevorzugung der häufig vorkommenden Klassen entsteht. Bei Inspektion der Klassifikationsergebnisse kommen die selten vorkommenden Klassen fast überhaupt nicht mehr vor. Daher können wir auch bei der Visualisierung der Ergebnisse teilweise sehr gute Ergebnisse erzielen: die kleinen Objektteile werden zwar in den Klassifikationsbildern nicht angezeigt, aber die häufig vorkommenden Klassen Gebäude, Fenster und Vegetation werden oftmals befriedigend dargestellt.

Dieses Kapitel haben wir mit einem pixelweisen Vergleich zwischen den Bildinterpretationen abgeschlossen. Dabei haben wir für manche Bilder eine Übereinstimmung von über 90% zwischen der Visualisierung der Targets der Regionen und der Visualisierung der Klassifikationsergebnisse durch das Bedingte Bayes-Netz vorgefunden. Dies zeigt uns das Potential unseres Verfahrens.

6. Zusammenfassung und Ausblick

6.1. Zusammenfassung der Arbeit und der Ergebnisse

In dieser Arbeit haben wir einen hierarchischen Ansatz für die Interpretation von Gebäudeaufnahmen vorgestellt. Das von uns entwickelte Verfahren ist universell einsetzbar, da es einen datengetriebenen konzipiert ist und damit sowohl terrestrische Gebäudeaufnahmen als auch Bilder aus anderen Perspektiven wie z. B. Luftbilder verarbeitet werden können. Die Vorgehensweise ist hierarchisch, weil wir die hierarchische Anordnung von Objektteilen auch für die Analyse des Bildes ausnutzen.

Die Regionenhierarchie verwenden wir zur Konstruktion eines Bedingten Bayes-Netzes, dessen Konzept und Realisierung zentraler Bestandteil unserer Arbeit ist. Die Knoten des Bedingten Bayes-Netzes stellen Zufallsvariablen dar, die jeweils die Zugehörigkeit einer Region zu vordefinierten Klassen beschreiben. Die Anordnung dieser Knoten im Bayes-Netz entspricht der Regionenhierarchie der segmentierten Regionen. Die Gesamtwahrscheinlichkeit für alle Zufallsvariablen zerlegen wir in Produkte von zwei Termen. Die Einen geben die Wahrscheinlichkeit für die Zugehörigkeit einer Region zu den Klassen an, die allein aus beobachteten regionenspezifischen Merkmalen bestimmt werden, die Anderen werden aus der Hierarchie der Regionen abgeleitet und geben die paarweisen Klassenzugehörigkeiten zwischen einer Region und der in der Hierarchie nach oben adjazenten Region an.

Für die Realisierung dieses Bedingten Bayes-Netzes integrieren wir sehr verschiedene Teilarbeiten. Zuerst analysieren wir viele dicht bei einander liegende Maßstabsebenen im Gauß'schen Skalenraum und segmentieren darauf aufbauend geometrisch präzise Regionen, die wir durch einen Regionen-Hierarchiegraphen anordnen. Später verwenden wir die daraus abgeleitete Struktur der nach einem Stabilitätskriterium ausgewählten Regionen zur Konstruktion des Bedingten Bayes-Netzes. Für diese stabilen Regionen bestimmen wir durch Vergleich mit manuell erstellten Annotationen die am besten passende der vorgegebenen Klassen und extrahieren geeignete regionenspezifische Merkmale. Diese Tupel aus Merkmalsvektor und Klassenlabel verwenden wir zum Lernen von Alternierenden Entscheidungsbäumen zur Klassifikation der Regionen, die Ergebnisse verwenden wir zur Initialisierung der Belegungen der Zufallsvariablen des Bedingten Bayes-Netzes. Die Selektion des besten Merkmals benutzen wir zum Lernen der Übergangswahrscheinlichkeiten im Bayes-Netz. Den effizienten Inferenz-Algorithmus von Pearl (1997) können wir wegen der baumartigen Struktur unseres Bayes-Netzes direkt verwenden.

Unsere vier Arbeitshypothesen haben wir auf insgesamt vier verschiedenen Datensätzen mit zusammen über 600 Bildern getestet. Dabei haben wir auch den Benchmark-Datensatz für die Interpretation von Fassadenbildern verwendet, der von Korč & Förstner (2009) zusammengestellt wurde. Ein weiterer Datensatz enthält Luftbildausschnitte, die anderen beiden zeigen terrestrische Perspektiven von Gebäuden bzw. entzerrte Fassadenfotos.

Wir haben in dieser Arbeit gezeigt, dass wir die stabilen Regionen unserer hierarchischen Bildsegmentierung gut zur Detektion von komplexen Objekten und ihren Bestandteilen verwenden können. In unseren Experimenten haben wir ein Detektionspotential von mehr als 90% für Gebäude, Himmel, Straßen und Vegetation, von mehr als 80% für Fenster und von

mehr als 70% für Autos und Türen festgestellt. Bei den anderen beiden verwendeten terrestrischen Datensätzen erzielen wir ähnliche Werte, lediglich die Möglichkeit, Fenster und Türen zu erkennen, sinkt um bis zu 20 Prozentpunkte. Auf den Luftbildausschnitten sind die Ergebnisse deutlich schlechter. Die Gründe für dieses Abschneiden sind an umfangreicherem Material zu untersuchen.

Wir haben ebenfalls gezeigt, dass die Relationen der Bestandteilshierarchie des Objekts in unsere Regionenhierarchie abgebildet werden. Bei den Benchmark- und anderen terrestrischen Aufnahmen erhalten wir eine Übereinstimmung von meistens 80%, bei den Luftbildausschnitten kommen wir dagegen nur auf etwas mehr als 60%. Dies liegt einerseits daran, dass weniger Objekte bzw. Objektteile durch unser Segmentierungsverfahren detektiert werden können, andererseits liegt es aber auch an der Beleuchtungssituation: Da viele Dächer schräg sind, was zu starken Reflektionsunterschieden von benachbarten Flächen führt. Da unsere Segmentierung auf einem Homogenitätskriterium basiert, kommt es hier verstärkt zur Verschmelzung von Regionen, die Objektteile verschiedener Objekte darstellen.

Bei der Merkmalsextraktionen berechnen wir 65 Merkmale für jede stabile Region, die wir entweder direkt aus den Pixeln der Region oder aus ihrer Umgebung herleiten. Bei der Inspektion der klassenspezifischen Verteilungen dieser Merkmale haben wir akzeptable Unterscheidungsmöglichkeiten bei vielen dieser Merkmale vorgefunden, wobei wir die Regionen zuvor nach Maßstabsebenen getrennt haben. Wir können belegen, dass sich der Einfluss der Merkmale bei der Klassifikation dieser Regionen verändert. In den unteren Maßstabsebenen verwendet unser Klassifikator verstärkt Farbmerkmale, in den oberen die Form beschreibende Merkmale. Zur Klassifikation haben wir die Methode der binären Alternierenden Entscheidungsbäume (ADTboost) auf den Mehrklassenfall verallgemeinert. Der schwierigste Punkt war dabei der Vergleich zwischen verschiedenen Klassifikationshypothesen, die bei jedem Boosting-Schritt evaluiert werden müssen. Obwohl wir nur achsenparallele Trennfunktionen verwenden, die nur ein originales Merkmal verwenden, erzielen wir etwas bessere Klassifikationsergebnisse als mit einer MAP-Klassifikation im optimalen LDA-Unterraum. Mit einer Erfolgsrate von 54.9% bei einer Berücksichtigung von sieben Klassen arbeitet unser ADTboost-Klassifikator auf dem Benchmark-Datensatz deutlich besser als der Zufall, aber mit diesem Ergebnis sind wir generell nicht zufrieden. Insbesondere auch deshalb, weil die drei besonders häufig vorkommenden Klassen Gebäude, Fenster und Vegetation durch den Klassifikator extrem bevorzugt werden. Die anderen Klassen kommen dagegen fast nicht vor. Unser Verfahren ist damit in der vorliegenden Konfiguration nicht geeignet, zuverlässig Regionen als Türen, Gauben, Balkone und Himmel etc. zu erkennen. Bei den anderen Datensätzen haben wir meistens mehr Klassen gegeben, was zu einer größeren Verwechslung zwischen den Klassen führt, weshalb auf den anderen Datensätzen die Klassifikationsrate leicht niedriger ausfällt.

Mit dieser Vorklassifikation durch ADTboost initialisieren wir die Belegung der Zufallsvariablen in unserem Bedingten Bayes-Netz. Die weiteren Übergangswahrscheinlichkeiten leiten wir aus der Struktur der Regionen ab, wozu wir das Auftreten der Klassen in den verschiedenen Maßstabsebenen in Abhängigkeit von beobachteten Merkmalsausprägungen zählen. Für die Inferenz der Information verwenden wir auf Grund der baumartigen Struktur unseres Bedingten Bayes-Netzes den Algorithmus von, der einen zweimaligen Durchlauf durch den Baum vorsieht: einen von unten nach oben sowie einen in umgekehrter Richtung. Nach dieser Weiterpropagierung der Information im Bayes-Netz haben wir noch einmal die bestmögliche Klassifikation für jede segmentierte Region bestimmt. Im Vergleich zur Vorklassifikation können wir uns um einige Prozentpunkte verbessern. Unsere Erwartungshaltung von einer Verbesserung von mehr als 10 Prozentpunkten oder mehr bleibt aber bei allen vier Datensätzen unerfüllt.

Abschließend haben wir unsere Bildinterpretation visualisiert. Dazu haben wir die Regionen

nach ihrer Maßstabszugehörigkeit und entsprechend der Klassenhierarchie angeordnet, damit kleine Objektteile nicht durch große Objekte verdecken. Bei einem pixelweisen Vergleich zwischen den Klassifikationszielen der Regionen, die wir durch Vergleich mit den manuellen Annotationen bestimmen, erzielen wir durchschnittlich eine Übereinstimmung von 70%. Viele Klassifikationsbilder liefern eine akzeptable Interpretation der Gebäudeszene und zeigen viele Objekte und wichtige ihrer Teile korrekt. Daraus schließen wir, dass unser Konzept des Bedingten Bayes-Netzes prinzipiell funktioniert. Bei einigen Aufnahmen sind die Ergebnisse der Bildinterpretation allerdings unbefriedigend, d. h. wir müssen einige Komponenten unseres Verfahrens noch verbessern. Da wir unseren Ansatz modular konzipiert haben, können wir einfach einzelne Arbeitsschritte erweitert oder durch bessere Methoden ausgetauscht werden.

6.2. Konzeptionelle Beiträge dieser Arbeit

Diese Arbeit hat viele verschiedene Themengebiete aufgegriffen und zu einem datengetriebenen, hierarchischen Ansatz für die Interpretation von Gebäudeaufnahmen zusammengesetzt. An dieser Stelle fassen wir noch einmal unsere wichtigsten inhaltlichen Beiträge zusammen.

- Die **hierarchische Segmentierung** unter Verwendung einer universellen Regionenhierarchie bildet das Fundament dieser Arbeit. Wir haben die Bildsegmentierung in verschiedenen Maßstabsebenen untersucht und eine präzise geometrische Beschreibung der Regionen durch das Verfolgen von Regionen durch den gesamten Maßstabsraum ermöglicht.

- Aus der Regionenhierarchie leiten wir ein **Bedingtes Bayes-Netz** ab, dessen Konzept und Realisierung den Kern dieser Arbeit darstellt. Wir postulieren, dass sich die Gesamtwahrscheinlichkeit der Zufallsvariablen des Bayes-Netzes durch das Produkt zweier Terme berechnen lässt. Einerseits benötigen wir Wahrscheinlichkeiten für das gemeinsame Auftreten zweier Klassen bezogen auf benachbarte Regionen der Regionenhierarchie. Diese bestimmen wir durch eine Analyse, wie häufig die Klassenpaare innerhalb der Trainingsmenge auftreten. Diese geben wir in Abhängigkeit eines beobachteten Merkmals an, das wir extra für diesen Zweck auswählen. Andererseits benötigen wir Wahrscheinlichkeiten für die Klassenzugehörigkeiten der Regionen in Abhängigkeit von beobachteten Merkmalen. Diese erhalten wir durch einen Klassifikator, dessen Parameter wir überwacht lernen. Da wir die Merkmale der Regionen beobachten, können wir alle anderen Zufallsvariablen des Bayes-Netzes unter dieser Bedingung formulieren. Entsprechend erhalten wir dann ein Bedingtes Bayes-Netz, dessen vereinfachte Struktur die von Bäumen aufweist. Damit ist auch die Inferenz innerhalb des Bayes-Netzes einfach durchzuführen.

- Die Erweiterung der **Klassifikation** mit Alternierenden Entscheidungsbäumen (ADTboost) auf den **Mehrklassenfall** ist ebenso ein wichtiger Bestandteil dieser Arbeit. Boosting-Verfahren haben sich in den vergangenen zehn Jahren stark entwickelt, wobei der allgemeine Fokus auf die Integration stärkerer Teilklassifikatoren liegt. Zudem wird in den meisten Verfahren der Mehrklassenfall auf den Zweiklassen zurückgeführt, was allerdings die Komplexität des Verfahrens und die Datenmenge erheblich vergrößert. Wir haben stattdessen die Bewertungsfunktion für verschiedene Hypothesen von Klassifikatoren in bestehenden Verfahren analysiert und eine Anpassung dieser Funktion für die Klassifikation mit mehreren Klassen aufgezeigt. Darüber hinaus bietet die hierar-

chische Anordnung der Teilklassifikatoren ein großes Potential für die Verbesserung von Klassifikationen.

6.3. Vorschläge für weiterführende Arbeiten

Wir möchten diese Arbeit mit einigen Ideen für weiterführende Arbeiten abschließen. Wir gliedern diese Liste inhaltlich und orientieren uns dabei an den drei großen Abschnitten zur Umsetzung unseres Konzepts: der Segmentierung, der Klassifikation und dem Graphischen Modell.

Segmentierung

Wir segmentieren unsere Regionen mittels des Wasserscheiden-Algorithmus. Dazu werden die Pixel zu homogenen Regionen gruppiert, ohne dass wir Wissen über die dargestellten Objekte oder die Beleuchtungsverhältnisse angeben. Eine mögliche Verbesserung wäre, dass Homogenitätskriterium für die Segmentierung der Wasserscheidenregionen zu verbessern. Gerade in Gebäudeszenen sind viele texturierte Objekte zu sehen. Die meisten Texturen wie bei den Blättern von Bäumen, oder bei den Dachschindeln und Backsteinen werden bei einer starken Glättung des Bildes auch farblich homogen, d. h. die entsprechenden Regionen existieren nur in höheren Maßstabsebenen. Außerdem stören Beleuchtungsunterschiede in den Bildern, insbesondere bei Schlagschatten. Dazu könnten die Arbeiten von Finlayson *et al.* (2006), Steinmetz *et al.* (2007), Sirmacek & Ünsalan (2008) und Arbel & Hel-Or (2011) beitragen. Problematisch könnte dabei sein, dass viele Schatten in Gebäudeszenen eine größere Fläche einnehmen als beispielweise in Gesichtern. Beide Erweiterungen, die farb- und texturbasierte Segmentierung sowie die Schattenentfernung, können zu noch besseren Segmentierungen der Objekte und zu einer besseren Regionenhierarchie führen.

Neuere Arbeiten wie z. B. Arbeláez *et al.* (2011) integrieren zusätzliche Kanteninformation und hochdimensionale Merkmalsvektoren in die auf Wasserscheiden basierte Segmentierung von Bildern. Eine hierarchisch strukturierte Bildeinteilung in Regionen ist dann durch sukzessives Verschmelzen der Regionen nach diversen Kriterien möglich. Dieser Mehraufwand hat möglicherweise keine größeren Auswirkungen auf die Laufzeit unseres Ansatzes, da der von uns entwickelte Segmentierungsansatz ohnehin jedes Bild in 41 Maßstabsebenen analysiert.

Unsere Segmentierung ist vollständig datengetrieben, was bei homogenen Objektstrukturen zu guten Ergebnissen, aber zu falschen Gruppierungen bei der Verschmelzung größerer Objektteile oder einander ähnlicher benachbarter Objekte führen kann. In Luftbildausschnitten sind die Dachteile und -aufbauten klein, so dass sie meistens in höheren Maßstabsebenen mit den umliegenden Dachregionen verschmelzen. Bei schrägen Dächern kommt es aber häufig vor, dass benachbarte Dachflächen das Sonnenlicht sehr unterschiedlich reflektieren, so dass hier die Dachflächen eher mit der Umgebung verschmelzen als dass sie zusammen das vollständige Dach ergeben. Bei terrestrischen Aufnahmen stellen große, dunkle Fenster in einer hellen Fassade ebenfalls ein Problem dar, da hier die Fensterkonturen trotz starker Glättung sichtbar bleiben können. Um solche Fehlsegmentierungen zu vermeiden, müsste Wissen über die Szene in die Segmentierung integriert werden, z. B. durch Hinzufügen von 3D-Information oder durch Anpassung des Homogenitätskriteriums, so dass rechteckige Strukturen bei der Segmentierung erhalten bleiben und nicht mit anderen freiförmigen Regionen verschmelzen. Erste Ergebnisse haben wir bereits durch Hinzufügen von geometrischer Information in die Segmentierung erzielt, die wir aus Bildsequenzen abgeleitet haben (Drauschke *et al.*, 2009).

Unsere Evaluation der Segmentierung basiert auf einem Vergleich der Regionen mit den manuell erstellten Annotationen. Trotz großer Sorgfalt sind sie weder einheitlich noch fehlerfrei. Bei der Bestimmung des bestmöglichen Klassenlabels für jede segmentierte Region haben wir einen Entscheidungsbaum verwendet, dessen Entscheidungen mittels eines Schwellwertvergleichs durchgeführt werden. Die Schwellwerte, den Vergleich und die Reihenfolge dieser Vergleiche haben wir manuell festgelegt und überprüft. Die Schwellwerte und die Reihenfolge der Vergleiche im Entscheidungsbaum sind bei einer Menge von vorgegebenen Vergleichen lern- und evaluierbar. Entsprechend kann auch die Visualisierung der Klassifikationsergebnisse besser durchgeführt werden. Wir haben lediglich die Maßstabsebenen und die Klassenhierarchie verwendet und Objekte gleicher Stufe in der Ontologie gemäß dem Klassifikationserfolg der zugehörigen Klassen berücksichtigt. Beide Teile wirken sich auf den pixelweisen Vergleich zwischen den manuell erstellten Annotationen und den Targets der segmentierten Regionen aus, bei dem wir derzeit nur eine durchschnittliche Übereinstimmung von 50% erzielen. Dieser geringe Wert offenbart noch ein deutliches Optimierungspotential unserer Arbeit.

Merkmale und Klassifikation

Die Liste der Merkmale ist nahezu beliebig erweiterbar. Wir haben uns auf 65 Merkmale beschränkt, um die Klassifikation mit vielen verschiedenen Methoden untersuchen zu können. Insbesondere haben wir diese Merkmale in die Arbeit von Yang *et al.* (2010) integriert, wo die Komplexität des Graphischen Modells deutlich höher ist als in unserem Fall. Wir können problemlos weitere Texturmerkmale integrieren, die wir bereits für die Untersuchung von Drauschke & Mayer (2010) implementiert haben. Wir können aber auch für jedes Pixel des Bildes die SIFT- bzw. HoG-Deskriptoren von Lowe (2004) bzw. Dalal & Triggs (2005) und diese bzgl. jeder Region mitteln. So können wir leicht hunderte neue Merkmale berechnen und damit unsere Merkmalsvektoren erweitern. Konsequenterweise hat diese Erweiterung der Merkmalsvektoren Auswirkungen auf die Laufzeit bei der Klassifikation mit Alternierenden Entscheidungsbäumen, da wir hier in jeder Iteration den besten einfachen Klassifikator auswählen müssen. Diese Anzahl der Berechnungen hängt bei unserer Realisierung des Klassifikators auch von der Anzahl der Merkmale ab.

Die Beschränkung der einfachen Klassifikatoren auf einzelne Merkmale hat zwei Gründe. Einerseits wollen wir die Komplexität von ADTboost eingrenzen, andererseits wollen wir anhand der Auswahl und Performanz dieser Klassifikatoren die Merkmalsrelevanz bestimmen. Alternativ kann man auch geeignete Untermengen des Merkmalsraums als Definitionsbereich eines einfachen Klassifikators zulassen. Dann müsste die deterministische Vorgehensweise bei der Auswahl des besten Klassifikators entweder randomisiert erfolgen oder durch Integration von Heuristiken analog zu (Pfahringer *et al.*, 2001) beschleunigt werden. Entsprechend müsste man dann die Merkmale auch durch zusätzliche Kriterien auswählen, wie es bei der Merkmalsselektion z. B. durch Ranking-Verfahren üblich ist (Guyon & Elisseeff, 2003).

Bei der Diskussion zu den Alternierenden Entscheidungsbäumen im Mehrklassenfall haben wir schon darauf hingewisen, dass die Entwicklung bei Adaboost zur Integration von relativ starken Klassifikatoren geführt hat. Es wäre somit interessant, ob und wie sich die Klassifikationsergebnisse von ADTboost bei Verwendung von Kernel-SVMs oder anderen starken Klassifikatoren wie die Random Forests oder Logistische Regression als einfache Klassifikatoren verbessern.

In unserem Boosting-Schema verwenden wir dieselben Gewichte für die Daten wir sie von Schapire & Singer (1999) und Freund & Mason (1999) vorgeschlagen wurde, d. h. als skalare Größen je Datensatz. Rätsch *et al.* (2001) dagegen gewichtet bei seiner Adaption der

Adaboost-Klassifikation die Daten zusätzlich nach Klassen, modelliert die Gewichte also als Vektor je Datensatz. Diese Vorgehensweise wird mit einer höheren Robustheit des Klassifikators begründet, weil so bei der Aktualisierung der Gewichte sehr ähnliche Klassen besonders berücksichtigt werden können. Bei der Klassifikation von Regionen, die Gebäude und Gebäudeteile darstellen, wurden wir mit genau diesem Problem konfrontiert.

Letztendlich müssten wir noch eine anschließende Klassifikation der künstlich eingefügten Klasse realisieren, in der wir alle zu gering vorkommenden Klassen zusammengefasst haben. Durch die starken Unterschiede zwischen den Instanzen dieser Klassen, sollte diese Klassifikation gute Ergebnisse liefern. An der Stelle könnte man auch überlegen, ob nicht mehr Klassen zusammengefasst werden sollten, um die Gewichtung zwischen den Klassen anzupassen: Auf dem Benchmark-Datensatz von Korč & Förstner (2009) stellen ungefähr 80% aller Regionen Objekte der Klassen Gebäude, Vegetation und Fenster dar.

Eine weitere Untersuchung sollte die Wahrscheinlichkeiten der Klassifikation thematisieren. Unsere Evaluationen basieren auf dem Vergleich der wahrscheinlichsten Klasse durch die Klassifikation und den Klassifikationszielen. Dabei haben wir nicht analysiert, wie gut die Abgrenzung zwischen der besten und z. B. der zweitbesten Klasse ist. Bei der Interpretation im Bayes-Netz ist uns das Verschwinden der selten vorkommenden Klassen bei der Ausgabe der Klassifikationsergebnisse aufgefallen. Somit sollte noch untersucht werden, ob dies an den Wahrscheinlichkeiten liegt, die der ADTboost-Klassifikator ausgibt. Dabei ist zu prüfen, ob und inwieweit generative Klassifikationsmodelle zu besseren Wahrscheinlichkeiten führen können als diskriminative.

Wir haben in unseren Experimenten drei Datensätze mit terrestrisch aufgenommenen Fassadenbildern und einen Luftbilddatensatz verwendet. Bartelsen & Mayer (2010) fotografieren die Gebäude mit unbemannten, flugfähigen Systemen (UAS). Dabei entstehen sowohl Fassadenbilder aus der Bodenperspektive und als Schrägluftbild als auch Dachansichten aus senkrechten sowie schrägen Aufnahmekonfigurationen. Wegen der verschiedenen Ansichten und vor allem wegen der sehr unterschiedlichen Beleuchtungsverhältnisse erwarten wir nicht, dass die in diesen Bildern segmentierten Regionen durch denselben Klassifikator, unabhängig davon ob ADTboost oder ein anderer, erfolgreich klassifiziert werden können. Daher sollte in einer nachfolgenden Arbeit auch die Möglichkeit zur Selektion der Klassifikator-Modelle untersucht werden. Hierbei ist zu beachten, dass der Übergang zwischen Fassadenaufnahmen und Senkrechtluftbildern durch die neuen Nahaufnahmen durch UAS fließend geworden ist.

Graphisches Modell und Bildinterpretation

In den vorherigen beiden Abschnitten haben wir eher Ideen präsentiert, wie man einzelne Teilschritte unseres Verfahrens verbessern kann. Hier stellen wir noch ein paar Möglichkeiten, unseren gesamten Ansatz zu verbessern.

Wir haben ein Bedingtes Bayes-Netz modelliert, in dem nur die Merkmale der Regionen beobachtet werden, während die anderen Zufallsvariablen unbeobachtet sind. Durch Integration von z. B. Fensterdetektoren könnte man die Vorklassifikation einzelner Regionen manipulieren, möglicherweise auch als Beobachtung interpretieren. Eine analoge Integration von Detektoren in ein Graphisches Modell haben Tu *et al.* (2005) bereits realisiert. Hier werden Schilder- und Gesichtsdetektoren mit den Ergebnissen aus der Interpretation mit einem Markoff-Zufallsfeld kombiniert, was die Interpretation der anderen Bildbereiche deutlich verbesserte. Wir könnten mögliche Detektionsergebnisse auf verschiedene Weisen in unser Bayes-Netz integrieren. Einerseits könnten wir die Wahrscheinlichkeit für die Zugehörigkeit zur detektierten Klasse bei den Regionen erhöhen, die mit den Detektionen überlappen, andererseits können wir dieselbe

Wahrscheinlichkeit bei den anderen Regionen verringern. Es wäre sicherlich eine Untersuchung wert, die Auswirkungen im Extremfall abzuschätzen, wenn die Wahrscheinlichkeiten für die Zugehörigkeit zur detektierten Klasse auf 1 bzw. auf 0 gesetzt werden.

Eine andere Folgearbeit zielt auf die Integration von Information aus den Nachbarschaften einer Maßstabsebene ein. Dazu müssten wir die Fokussierung auf die stabilen Regionen fallen lassen und mit Bildpartitionierungen in ausgewählten Maßstabsebenen arbeiten, siehe (Yang *et al.*, 2010). Dieser Ansatz integriert nachbarschaftliche und hierarchische Relationen in ein Graphisches Modell. Damit wird das Lernen von Abhängigkeiten zwischen den Merkmalen und den Klassen sehr aufwändig und konnte, unseres Wissens nach, noch nicht erfolgreich abgeschlossen werden. Wenn man nur die Relationen zwischen Regionen einer Maßstabsebene betrachtet, so lässt sich ein Bedingtes Markoff-Zufallsfeld formulieren, dass die Nachbarschaftsinformationen bei der Szeneninterpretation berücksichtigt. Wenn man nur die hierarchischen Relationen betrachtet, kann unser Bedingtes Bayes-Netz angewendet werden. Wechselt man nun iterativ zwischen der Inferenz in Markoff-Zufallsfeldern und der Inferenz im Bayes-Netz, dann könnte man ein effizientes und erfolgreiches Verfahren zur Bildinterpretation erhalten. Dieser Ansatz könnte entweder nach einer festen Anzahl von Schritten beendet werden, oder wenn sich die Klassifikationsergebnisse der Regionen nicht mehr ändern, d. h. die Bildinterpretation einen stabilen Zustand eingenommen hat.

Wir haben unsere Arbeit stark auf die Segmentierung und Interpretation von Gebäudebilder fokussiert, obwohl wir nur sehr wenige Annahmen bzgl. dieser Anwendung getroffen haben. Beispielsweise haben wir Wasserscheiden zur Segmentierung der Bilder verwendet, weil Gebäude und ihre Teile i. d. R. starke Kanten zeigen und daher die Regionen gut zu ihnen passen. Es sollte noch untersucht werden, ob das Bedingte Bayes-Netz im Allgemeinen bzw. welche seiner Komponenten für die Interpretation anderer Bilder verwendet werden können. Anerkannte Benchmark-Datensätze sind der MSRC von Shotton *et al.* (2006) sowie ImageNet von Deng *et al.* (2009). Allerdings enthalten die Annotationen dieser Datensätze keine überlappenden Klassen, d. h. es gibt keine Bestandteilshierarchie von Objekten. Entsprechend müsste die Target-Zuordnung angepasst werden, die Regionenhierarchie spiegelt dann keine Objektstruktur wider, sondern ermöglicht eine Mehrskalen-Analyse analog zur Arbeit von Lim *et al.* (2009).

Fazit der Arbeit

Als Fazit dieser Arbeit sehen wir, dass sich Bedingte Bayes-Netze für die Integration von Kontext bei der Klassifikation von Bildregionen eignen. Allerdings hängt der Erfolg stark von der Performanz des verwendeten Klassifikators ab. Wir erzielen zufriedenstellende Ergebnisse in Bezug auf die Erkennung von häufig auftretenden Klassen. Allerdings haben wir unseren Ansatz auch mit der Erkennung kleinerer Details motiviert, diesbzgl. versagt unsere Bildinterpretation.

Die Komponenten unseres Verfahrens, d. h. die hierarchische Segmentierung, die Merkmalsextraktion und die Klassifikation der Regionen können erweitert oder ausgetauscht werden, ohne dass sich das auf die Formulierung des Gesamtverfahrens des Bedingten Bayes-Netzes auswirkt. Damit ist das Verfahren leicht für andere Anwendungen adaptierbar.

A. Anhang

A.1. Charakterisierung der extrahierten Merkmale

In Kap. 4.1 haben wir 65 verschiedene Merkmale vorgestellt, die wir bei unserer Klassifikation der stabilen Regionen verwenden. Hier listen wir diese Merkmale noch einmal auf und geben zusätzlich drei charakteristische Werte für diese Merkmale an, die wir aus den segmentierten Regionen auf dem Benchmark-Datensatz `terra-1` herleiten. Der minimale und der maximale Wert des Merkmals geben Auskunft über den Wertebereich des Merkmals, zudem geben das arithmische Mittel der Merkmalsausprägungen an.

Tabelle A.1.: Auflistung und Charakterisierung der 65 verwendeten regionenspezifischen Merkmale

Index	Name	Min	Mean	Max
1	Anzahl der Komponenten	1	1.0466	54
2	Anzahl der Löcher	0	0.1523	95
3	Euler-Charakteristik	-94	0.8944	54
4	Fläche	1	3009.6099	1280833
5	Umfang	4	183.033	12423
6	Formfaktor	1.2732	2.8806	129.2796
7	Höhe der Bounding Box	1	29.1268	1533
8	Breite der Bounding Box	1	39.0548	1256
9	Verhältnis von Fläche der Region zu Fläche der Bounding Box	0.001	0.555	0.9634
10	relative Höhe des Mittelpunkts der Bounding Box	0	0.4507	0.9982
11	relative Breite des Mittelpunkts der Bounding Box	0.0017	0.5064	1
12	mittlerer Rotwert	0	91.9723	255
13	mittlerer Grünwert	0	94.7893	253.8148
14	mittlerer Blauwert	0	86.6403	255
15	mittlerer Hue-Wert	0	0.24167	0.9999
16	Streuung der Rotwerte	0	21.1986	100.318
17	Streuung der Grünwerte	0	20.9588	99.4736
18	Streuung der Blauwerte	0	21.2382	100.9156
19	Differenz der mittleren Rotwerte zwischen der Region und ihrer Umgebung	-217.2767	0.9560	225.8024
20	Differenz der mittleren Grünwerte zwischen der Region und ihrer Umgebung	-214.127	0.5948	225.8024
21	Differenz der mittleren Blauwerte zwischen der Region und ihrer Umgebung	-211.9259	0.646	225.8024
22	Differenz der mittleren Hue-Werte zwischen der Region und ihrer Umgebung	-0.4999	-0.028553	0.5
23	Differenz der Streuung der Rotwerte zwischen der Region und ihrer Umgebung	-53.349	18.1814	93.453
24	Differenz der Streuung der Grünwerte zwischen der Region und ihrer Umgebung	-53.3776	18.2407	94.4454

Index	Name	Min	Mean	Max
25	Differenz der Streuung der Blauwerte zwischen der Region und ihrer Umgebung	-49.3403	18.6273	100.9366
26	Differenz zwischen dem höchsten und dem zweithöchsten Wert des normalisierten Gradientenhistogramms des roten Kanals	0	0.1594	1
27	Differenz zwischen dem höchsten und dem dritthöchsten Wert des normalisierten Gradientenhistogramms des roten Kanals	0.0004	0.2709	1
28	Entropie des normalisierten Gradientenhistogramms des roten Kanals	0	1.5218	2.0788
29	Differenz zwischen der Entropie des normalisierten Gradientenhistogramms der Region und der Entropie des normalisierten Gradientenhistogramms ihrer Umgebung des roten Kanals	-1.3257	0.21	2.0593
30	Differenz zwischen dem höchsten und dem zweithöchsten Wert des normalisierten Gradientenhistogramms des grünen Kanals	0	0.1582	1
31	Differenz zwischen dem höchsten und dem dritthöchsten Wert des normalisierten Gradientenhistogramms des grünen Kanals	0.0002	0.2707	1
32	Entropie des normalisierten Gradientenhistogramms des grünen Kanals	0	1.5199	2.0787
33	Differenz zwischen der Entropie des normalisierten Gradientenhistogramms der Region und der Entropie des normalisierten Gradientenhistogramms ihrer Umgebung des grünen Kanals	-1.2919	0.2077	2.0697
34	Differenz zwischen dem höchsten und dem zweithöchsten Wert des normalisierten Gradientenhistogramms des blauen Kanals	0	0.161	1
35	Differenz zwischen dem höchsten und dem dritthöchsten Wert des normalisierten Gradientenhistogramms des blauen Kanals	0	0.2719	1
36	Entropie des normalisierten Gradientenhistogramms des blauen Kanals	0	1.5209	2.0794
37	Differenz zwischen der Entropie des normalisierten Gradientenhistogramms der Region und der Entropie des normalisierten Gradientenhistogramms ihrer Umgebung des blauen Kanals	-1.1034	0.20834	2.0692
38	Element der Momentenmatrix (Nebendiagonalelement)	-10125.4264	-4.2328	9186.3443
39	Element der Momentenmatrix (1. Element auf der Diagonale)	0	226.1668	40269.9126
40	Element der Momentenmatrix (2. Element auf der Diagonale)	0	648.2347	49151.9167

Index	Name	Min	Mean	Max
41	Größerer Eigenwert der Momentenmatrix	0	728.5818	49274.2209
42	Kleinerer Eigenwert der Momentenmatrix	0	145.8197	27451.5917
43	Verhältnis der beiden Eigenwerte (kleiner zu größerem)	0	0.31859	1
44	Orientierung der Hauptachse (in rad)	-1.5708	0.030611	1.5708
45	Verhältnis zwischen der Anzahl der Pixel im Schnitt und der Anzahl der Pixel in der Vereinigung der Region mit dem generalisierten Viereck	0.0028245	0.82628	1
47	Erster Winkel des generalisierten Vierecks	0.91919	1.771	3.1339
46	Zweiter Winkel des generalisierten Vierecks	0.024995	1.3724	2.9275
49	Dritter Winkel des generalisierten Vierecks	0.75838	1.7648	3.1133
48	Vierter Winkel des generalisierten Vierecks	0.015281	1.375	2.8706
50	Texturmerkmal (Filter 1. Reihe, 1. Spalte)	2.0869	91.4775	251.7715
51	Texturmerkmal (Filter 1. Reihe, 2. Spalte)	-75.3183	-0.4686	48.3197
52	Texturmerkmal (Filter 1. Reihe, 3. Spalte)	-42.1789	-0.3775	34.9629
53	Texturmerkmal (Filter 1. Reihe, 4. Spalte)	-29.6947	0.2211	44.181
54	Texturmerkmal (Filter 2. Reihe, 1. Spalte)	-57.5764	-0.4896	77.75
55	Texturmerkmal (Filter 2. Reihe, 2. Spalte)	-52.2742	0.0243	45.5398
56	Texturmerkmal (Filter 2. Reihe, 3. Spalte)	-15.1134	0.0256	19.4213
57	Texturmerkmal (Filter 2. Reihe, 4. Spalte)	-22.0522	-0.0239	19.5358
58	Texturmerkmal (Filter 3. Reihe, 1. Spalte)	-43.6381	-0.311	44.6036
59	Texturmerkmal (Filter 3. Reihe, 2. Spalte)	-18.7638	0.0303	24.3568
60	Texturmerkmal (Filter 3. Reihe, 3. Spalte)	-11.5	0.0299	15.3736
61	Texturmerkmal (Filter 3. Reihe, 4. Spalte)	-12.5374	-0.023	9.5
62	Texturmerkmal (Filter 4. Reihe, 1. Spalte)	-37.2336	0.1115	43.5031
63	Texturmerkmal (Filter 4. Reihe, 2. Spalte)	-28.658	-0.0248	28.3933
64	Texturmerkmal (Filter 4. Reihe, 3. Spalte)	-14.3339	-0.021	13.75
65	Texturmerkmal (Filter 4. Reihe, 4. Spalte)	-18.9409	0.0094	13.1579

A.2. Datensatz von Fukunaga

Der Datensatz von Fukunaga (1972) besteht aus vier normalverteilten Klassen im 8-dimensionalen Raum. Wir haben die Daten dieser Klassen in Abb. 4.17 visualisiert. Dabei haben die Punkte der ersten Klasse blau gefärbt, die der zweiten Klasse rot, die der dritten Klasse grün und die der vierten Klasse schwarz. Die Klassen werden durch die beiden Parameter μ und Σ repräsentiert. Die Mittelwerte $\boldsymbol{\mu}$ geben wir zeilenweise je Klasse in Gl. A.1 an, die Kovarianzmatrizen Σ_i haben wir mit dem Klassenindex versehen und stellen sie in den Gl. A.2 bis A.5 dar.

$$\boldsymbol{\mu} = \begin{pmatrix} 7.825 & 6.750 & 5.835 & 8.525 & 6.625 & 7.065 & 7.865 & 4.435 \\ 5.760 & 5.715 & 5.705 & 4.150 & 6.225 & 6.960 & 6.750 & 3.910 \\ 6.610 & 5.060 & 5.980 & 3.975 & 9.020 & 14.685 & 10.640 & 4.175 \\ 6.120 & 6.285 & 5.850 & 4.365 & 6.340 & 4.675 & 6.260 & 4.440 \end{pmatrix} \tag{A.1}$$

$$\Sigma_1 = \begin{pmatrix} 1.034 & 1.281 & 0.351 & -0.293 & 0.098 & 0.301 & 0.141 & 1.336 \\ 1.281 & 1.967 & 0.664 & -0.219 & 0.259 & 0.556 & 0.276 & 2.094 \\ 0.351 & 0.664 & 7.138 & 1.192 & 2.726 & 1.116 & 0.678 & 2.097 \\ -0.293 & -0.219 & 1.192 & 2.269 & 1.367 & 0.146 & 0.201 & -0.308 \\ 0.098 & 0.259 & 2.726 & 1.367 & 5.727 & 1.280 & 0.933 & 2.107 \\ 0.301 & 0.556 & 1.116 & 0.146 & 1.280 & 2.941 & 1.949 & 2.197 \\ 0.141 & 0.276 & 0.678 & 0.201 & 0.933 & 1.949 & 1.577 & 1.229 \\ 1.336 & 2.094 & 2.097 & -0.308 & 2.107 & 2.197 & 1.229 & 6.606 \end{pmatrix} \tag{A.2}$$

$$\Sigma_2 = \begin{pmatrix} 4.792 & 4.417 & 4.244 & 2.406 & 1.798 & 0.790 & 0.785 & 2.993 \\ 4.417 & 5.074 & 4.636 & 2.798 & 1.824 & 0.639 & 0.644 & 2.799 \\ 4.244 & 4.636 & 5.428 & 3.224 & 2.111 & 0.903 & 1.131 & 2.943 \\ 2.406 & 2.798 & 3.224 & 5.287 & 3.006 & 1.326 & 1.897 & 2.648 \\ 1.798 & 1.824 & 2.111 & 3.006 & 3.574 & 2.229 & 2.471 & 1.915 \\ 0.790 & 0.639 & 0.903 & 1.326 & 2.229 & 4.008 & 2.405 & 1.106 \\ 0.785 & 0.644 & 1.131 & 1.897 & 2.471 & 2.405 & 4.507 & 1.727 \\ 2.993 & 2.799 & 2.943 & 2.648 & 1.915 & 1.106 & 1.727 & 3.972 \end{pmatrix} \tag{A.3}$$

$$\Sigma_3 = \begin{pmatrix} 1.638 & 2.153 & 1.482 & 1.695 & -0.557 & -2.443 & -0.710 & 1.983 \\ 2.153 & 3.596 & 2.461 & 2.436 & -0.591 & -3.711 & -0.493 & 2.434 \\ 1.482 & 2.461 & 2.500 & 2.834 & -0.665 & -2.621 & 0.248 & 1.738 \\ 1.695 & 2.436 & 2.834 & 4.704 & -0.629 & -2.913 & 0.576 & 2.471 \\ -0.557 & -0.591 & -0.665 & -0.629 & 19.000 & 0.896 & 8.622 & -0.254 \\ -2.443 & -3.711 & -2.621 & -2.913 & 0.896 & 5.856 & 1.357 & -2.915 \\ -0.710 & -0.493 & 0.248 & 0.576 & 8.622 & 1.357 & 20.800 & -0.622 \\ 1.983 & 2.434 & 1.738 & 2.471 & -0.254 & -2.915 & -0.622 & 3.214 \end{pmatrix} \tag{A.4}$$

$$\Sigma_4 = \begin{pmatrix} 5.116 & 4.736 & 4.058 & 1.821 & 1.109 & 1.289 & 1.029 & 2.232 \\ 4.737 & 5.684 & 4.523 & 2.311 & 1.273 & 1.328 & 1.151 & 2.425 \\ 4.058 & 4.523 & 6.117 & 2.525 & 1.321 & 1.501 & 1.274 & 2.191 \\ 1.821 & 2.311 & 2.525 & 4.432 & 2.481 & 2.179 & 1.080 & 1.784 \\ 1.109 & 1.273 & 1.321 & 2.481 & 2.134 & 2.325 & 1.017 & 1.030 \\ 1.289 & 1.328 & 1.501 & 2.179 & 2.325 & 4.099 & 2.019 & 1.803 \\ 1.029 & 1.151 & 1.274 & 1.080 & 1.017 & 2.019 & 1.872 & 2.081 \\ 2.232 & 2.425 & 2.191 & 1.784 & 1.030 & 1.803 & 2.081 & 3.806 \end{pmatrix} \tag{A.5}$$

A.3. Konfusionsmatrizen der Klassifikation durch das Bedingte Bayes-Netz

In Kap. 5.3 haben wir auf die Angabe der Konfusionsmatrix zur Dokumentation des Klassifikationserfolgs auf dem Benchmark-Datensatz `terra-1` verzichtet, um sie hier zusammen mit den Konfusionsmatrizen der drei anderen Datensätze anzugeben. Die Tab. A.2 spiegelt den Klassifikationserfolg von Datensatz `terra-1` wider, Tab. A.3 den von Datensatz `terra-2`, Tab. A.4 den von Datensatz `terra-3` und Tab. A.5.

Tabelle A.2.: Konfusionsmatrix und relative Erfolge der Klassifikation mittels Bedingten Bayes-Netzes unter Verwendung der Klassifikation durch ADTboost für Datensatz `terra-1` mit den $K = 7$ Klassen für Gebäude, Gebäudeteil-Mixturen, Autos, Vegetation, Fenster, Hintergrund und die eine zusammengelegte Klasse für den Rest und bei Verwendung von 131 060 Regionen. Die fett geschriebenen Werte auf der Hauptdiagonalen zeigen die Erkennungsraten der jeweiligen Klassen an. Insgesamt werden 79 476 Regionen korrekt klassifiziert, was einem Erfolg von 60.6% entspricht. Jede Zeile zeigt die tatsächliche Klasse $\widetilde{x}_m$ der Regionen an, jede Spalte das Klassifikationsergebnis $\widehat{x}_m$.

Klasse	others	building	car	vegetation	window	building-mixture	background
others	**2 835**	7 793	15	801	505	0	0
	23.7%	65.2%	0.1%	6.7%	4.2%	0.0%	0.0%
building	151	**35 011**	0	2 127	4 531	5	3
	0.4%	**83.7**%	0.0%	5.1%	10.8%	0.0%	0.0%
car	357	3 817	**16**	445	200	0	1
	7.4%	78.9%	**0.3**%	9.2%	4.1%	0.0%	0.0%
vegetation	238	6 474	2	**24 498**	920	0	0
	0.7%	20.1%	0.0%	**76.2**%	2.9%	0.0%	0.0%
window	7	13 400	0	549	**17 115**	2	0
	0.0%	43.1%	0.0%	1.8%	**55.1**%	0.0%	0.0%
building-mixture	0	2 551	0	103	1 316	**1**	0
	0.0%	64.2%	0.0%	2.6%	33.1%	**0.0**%	0.0%
background	347	3 112	1	1 701	110	0	**0**
	6.6%	59.0%	0.0%	32.3%	2.1%	0.0%	**0.0**%

Tabelle A.3.: Konfusionsmatrix und relative Erfolge der Klassifikation mittels Bedingten Bayes-Netzes unter Verwendung der Klassifikation durch ADTboost für Datensatz `terra-2` mit den $K = 8$ Klassen für Balkone, Himmel, Fenster, Vegetation, Fassaden-Mixturen, Gebäude-Mixturen, Dach-Mixturen und die eine zusammengelegte Klasse für den Rest und bei Verwendung von 145 534 Regionen. Die fett geschriebenen Werte auf der Hauptdiagonalen zeigen die Erkennungsraten der jeweiligen Klassen an. Insgesamt werden 71 361 Regionen korrekt klassifiziert, was einem Erfolg von 49.0% entspricht. Jede Zeile zeigt die tatsächliche Klasse $\widetilde{x}_m$ der Regionen an, jede Spalte das Klassifikationsergebnis $\widehat{x}_m$.

Klasse	others	balcony	sky	window	vegetation	facade	building-mix	roof-mix
others	**11 713**	0	171	7 280	250	8 506	0	27
	41.9%	0.0%	0.6%	26.0%	0.9%	30.4%	0.0%	0.1%
balcony	701	**0**	0	2 067	30	2208	0	3
	14.0%	**0.0**%	0.0%	41.3%	0.6%	44.1%	0.0%	0.1%
sky	364	0	**1 090**	1 579	326	1 667	0	143
	7.0%	0.0%	**21.1**%	30.5%	6.3%	32.2%	0.0%	2.8%
window	3 214	2	4	**33 865**	43	7 522	2	16
	7.2%	0.0%	0.0%	**75.8**%	0.1%	16.8%	0.0%	0.0%
vegetation	2 189	0	256	1 885	**1 717**	3 027	2	65
	23.9%	0.0%	2.8%	20.6%	**18.8**%	33.1%	0.0%	0.7%
facade	7 850	0	35	9 859	101	**22 914**	7	24
	19.2%	0.0%	0.1%	24.2%	0.2%	**56.2**%	0.0%	0.0%
building-mixture	959	0	23	1 351	121	2 608	**0**	18
	18.9%	0.0%	0.5%	26.6%	2.4%	51.3%	**0.0**%	0.4%
roof-mixture	487	0	42	2 343	121	4 642	0	**62**
	6.3%	0.0%	0.5%	30.4%	1.6%	60.3%	0.0%	**0.8**%

Tabelle A.4.: Konfusionsmatrix und relative Erfolge der Klassifikation mittels Bedingten Bayes-Netzes unter Verwendung der Klassifikation durch ADTboost für Datensatz `terra-3` mit den $K = 8$ Klassen für Boden, Vegetation, Fenster, Himmel, Fassaden-Teile, Dach-Teile, Hintergrund und die eine zusammengelegte Klasse für den Rest und bei Verwendung von 230 848 Regionen. Die fett geschriebenen Werte auf der Hauptdiagonalen zeigen die Erkennungsraten der jeweiligen Klassen an. Insgesamt werden 115 989 Regionen korrekt klassifiziert, was einem Erfolg von 50.2% entspricht. Jede Zeile zeigt die tatsächliche Klasse $\widetilde{x}_m$ der Regionen an, jede Spalte das Klassifikationsergebnis $\widehat{x}_m$.

Klasse	others	ground	vegetation	window	sky	facade	roof-mix	background
others	**1 902**	1	251	8 582	86	17 116	48	7
	6.8%	0.0%	0.9%	30.7%	0.3%	61.1%	0.2%	0.0%
ground	117	**4**	735	793	0	6 179	0	5
	1.5%	**0.1**%	9.4%	10.1%	0.0%	78.9%	0.0%	0.1%
vegetation	105	1	**8 311**	6 411	550	9 261	214	25
	0.4%	0.0%	**33.4**%	25.8%	2.2%	37.2%	0.9%	0.1%
window	43	0	289	**47 255**	60	16 136	9	6
	0.1%	0.0%	0.5%	**74.1**%	0.1%	25.3%	0.0%	0.0%
sky	85	0	394	2 473	**4 165**	2 847	114	76
	0.8%	0.0%	3.9%	24.4%	**41.0**%	28.0%	1.1%	0.7%
facade	412	0	613	19 915	82	**53 355**	35	34
	0.5%	0.0%	0.8%	26.8%	0.1%	**71.7**%	0.0%	0.0%
roof-mixture	86	0	328	5 999	177	7 670	**331**	30
	0.6%	0.0%	2.2%	41.0%	1.2%	52.4%	**2.3**%	0.2%
background	14	0	166	1 982	416	3 830	47	**666**
	0.2%	0.0%	2.3%	27.8%	5.8%	53.8%	0.7%	**9.4**%

Tabelle A.5.: Konfusionsmatrix und relative Erfolge der Klassifikation mittels Bedingten Bayes-Netzes unter Verwendung der Klassifikation durch ADTboost für Datensatz `aerial` mit den $K = 6$ Klassen für Autos, Gebäude, Gebäudeteil-Mixturen, Dachschindeln, Hintergrund und die eine zusammengelegte Klasse für den Rest und bei Verwendung von 64 339 Regionen. Die fett geschriebenen Werte auf der Hauptdiagonalen zeigen die Erkennungsraten der jeweiligen Klassen an. Insgesamt werden 34 610 Regionen korrekt klassifiziert, was einem Erfolg von 53.8% entspricht. Jede Zeile zeigt die tatsächliche Klasse $\widetilde{x}_m$ der Regionen an, jede Spalte das Klassifikationsergebnis $\widehat{x}_m$.

Klasse	`others`	`car`	`building`	`mixture`	`roof`	`background`
`others`	**229**	374	331	0	3 811	1 855
	3.5%	5.7%	5.0%	0.0%	57.7%	28.1%
`car`	54	**1 628**	400	0	2 201	1 588
	0.9%	**27.7**%	6.8%	0.0%	37.5%	27.0%
`building`	53	389	**1 064**	1	4 390	2 308
	0.6%	4.7%	**13.0**%	0.0%	53.5%	18.1%
`mixture`	11	52	151	**0**	998	751
	0.6%	2.6%	7.7%	**0.0**%	50.8%	38.3%
`roof`	78	540	598	0	**11 873**	6 270
	0.4%	2.8%	3.1%	0.0%	**61.3**%	32.4%
`background`	42	169	175	1	2 138	**19 816**
	0.2%	0.8%	0.8%	0.0%	9.6%	**88.7**%

Tabellenverzeichnis

Abbildungsverzeichnis

Literaturverzeichnis

Agarwal, S., Furukawa, Y., Snavely, N., Curless, B., Seitz, S. M., & Szeliski, R. 2010. Reconstructing Rome. *IEEE Computer*, **43**(6), 40–47.

Ali, H., Seifert, C., Jindal, N., Paletta, L., & Paar, G. 2007. Window Detection in Facades. *Pages 837–842 of: ICIAP 2007.*

Allwein, E. L., Schapire, R. E., & Singer, Y. 2000. Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. *JMLR*, **1**, 113–141.

Arbel, E., & Hel-Or, H. 2011. Shadow Removal using Intensity Surfaces and Texture Anchor Points. *PAMI*, **33**(6), 1202–1216.

Arbeláez, P., Maire, M., Fowlkes, C., & Malik, J. 2009. From Contours to Regions: An Empirical Evaluation. *Pages 2294 – 2301 of: CVPR.*

Arbeláez, P., Maire, M., Fowlkes, C., & Malik, J. 2011. Contour Detection and Hierarchical Image Segmentation. *PAMI*, **33**(5), 898–916.

Asuncion, A., & Newman, D. J. 2007. *UCI Machine Learning Repository.*

Ballard, D. H., & Brown, C. M. 1982. *Computer Vision.* Prentice Hall.

Bartelsen, J., & Mayer, H. 2010. Orientation of Image Sequences Acquired from UAVs and with GPS Cameras. *Land and Information Science*, **70**(3), 151–159.

Becker, S. 2010. *Automatische Ableitung und Anwendung von Regeln für die Rekonstruktion von Fassaden aus heterogenen Sensordaten.* Ph.D. thesis, Universität Stuttgart.

Berg, A. C., Grabler, F., & Malik, J. 2007. Parsing Images of Architectural Scenes. *In: ICCV.*

Bergholm, F. 1987. Edge Focusing. *PAMI*, **9**(6), 726–741.

Biedermann, I. 1987. Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, **94**(2), 115–147.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning.* Springer.

Borenstein, E., & Ullman, S. 2004. Learning to Segment. *Pages (III) 315–328 of: ECCV.* LNCS 3023.

Boyd, S., & Vandenberghe, L. 2008. *Convex Optimization.* 6th edn. Cambridge University Press.

Breiman, L. 2001. Random Forests. *Machine Learning*, **45**(1), 5–32.

Burochin, J.-P., Tournaire, O., & Paparoditis, N. 2009. An Unsupervised Hierarchical Segmentation of a Facade Building Image in Elementary 2D-Models. *Pages 223–228 of: CMRT'09.*

Čech, J., & Šára, R. 2009. Languages for Constrained Binary Segmentation based on Maximum Aposteriori Probability Labeling. *International Journal of Imaging and Technology*, **19**(2), 66–99.

Dalal, N., & Triggs, B. 2005. Histograms of Oriented Gradients for Human Detection. *Pages I: 886–893 of: CVPR.*

De Comité, F., Gilleron, R., & Tommasi, M. 2001. Learning Multi-label Alternating Decision Trees and Applications. *Pages 195–210 of: CAP'01.*

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. 2009. ImageNet: A Large-Scale Hierarchical Image Database. *Pages 248–255 of: CVPR.*

Dick, A. R. 2001. *Modelling and Interpretation of Architecture from Several Images.* Ph.D. thesis, University of Cambridge.

Dörschlag, D., Gröger, G., & Plümer, L. 2008. Über die schrittweise Erstellung und Verfeinerung von Modellhypothesen für Gebäude. *PFG*, **2008**(3), 157–164.

Douglas, D. H., & Peucker, T. K. 1973. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *Canadian Cartographer*, **10**(2), 112–122.

Drauschke, M. 2009. An Irregular Pyramid for Multi-Scale Analysis of Objects and their Parts. *Pages 293–303 of: GbR'09.* LNCS 5534.

Drauschke, M., & Förstner, W. 2008. Comparison of Adaboost and ADTboost for Feature Subset Selection. *Pages 113–122 of: PRIS.*

Drauschke, M., & Förstner, W. 2011. A Bayesian Approach for Scene Interpretation with Integrated Hierarchical Structure. *Pages 1–10 of: DAGM.* LNCS 6835.

Drauschke, M., & Mayer, H. 2010. Evaluation of Texture Energies for Classification of Facade Images. *Pages 257–262 of: PCV'10.* IAPRS 38 (3A).

Drauschke, M., Schuster, H.-F., & Förstner, W. 2006a. Detectability of Buildings in Aerial Images over Scale Space. *Pages 7–12 of: PCV'06.* IAPRS 36 (3).

Drauschke, M., Schuster, H.-F., & Förstner, W. 2006b. Stabilität von Regionen im Skalenraum. *Pages 29–36 of: Publikationen der DGPF: Geoinformatik und Erdbeobachtung*, vol. 15.

Drauschke, M., Roscher, R., Läbe, T., & Förstner, W. 2009. Improving Image Segmentation using Multiple View Analysis. *In: CMRT'09.*

Duda, R. O., Hart, P. E., & Stork, D. G. 2001. *Pattern Classification.* John Wiley and Sons.

Eibl, G., & Pfeiffer, K.-P. 2005. Multiclass Boosting for Weak Classifiers. *Journal of Machine Learning Research*, **6**, 189–210.

Epshtein, B., & Ullman, S. 2005. Feature Hierarchies for Object Classification. *Pages 220–227 of: ICCV.*

Epshtein, B., & Ullman, S. 2007. Semantic Hierarchies for Recognizing Objects and Parts. *Pages 1–8 of: CVPR.*

Fairfield, J. 1992. Toboggan Contrast Enhancement. *Pages 282–292 of: Applications of Artificial Intelligence X: Machine Vision and Robotics.* SPIE 1708.

Fei-Fei, L., Fergus, R., & Petrona, P. 2006. One-Shot Learning of Object Categories. *PAMI*, **28**(4), 594–611.

Finlayson, G. D., Hordley, S. D., Lu, C., & Drew, M. S. 2006. On the Removal of Shadows from Images. *PAMI*, **28**(1), 59–68.

Fischer, A. 2005. *Automatische Gebäuderekonstruktion mittels parametrisierter Komponenten.* Ph.D. thesis, Universität Bonn.

Fischer, A., Kolbe, T. H., Lang, F., Cremers, A. B., Förstner, W., Plümer, L., & Steinhage, V. 1998. Extracting Buildings from Aerial Images using Hierarchical Aggregation in 2D and 3D. *CVIU*, **72**(2), 185–203.

Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. 1996. *Computer Graphics: Principles and Practice.* 2nd ed. in C edn. Addison-Wesley.

Förstner, W. 1994. A Framework for Low Level Feature Extraction. *Pages II: 383–394 of: ECCV.* LNCS 801.

Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., & Pollefeys, M. 2010. Building Rome on a Cloudless Day. *Pages 368–381 of: ECCV.* LNCS 6314.

Franc, V., & Hlaváč, V. 2004. *Statistical Pattern Recognition Toolbox for Matlab.* Tech. rept. CMP, Czech Technical University, http://cmp.felk.cvut.cz/cmp/software/stprtool/.

Freund, Y., & Mason, L. 1999. The Alternating Decision Tree Learning Algorithm. *Pages 124–133 of: ICML'99.*

Freund, Y., & Schapire, R. E. 1996. Experiments with a New Boosting Algorithm. *Pages 148–156 of: ICML'96.*

Fuchs, C. 1998. *Extraktion polymorpher Bildstrukturen und ihre topologische und geometrische Gruppierung.* Ph.D. thesis, Rheinische Friedrich-Wilhelms-Universität Bonn.

Fukunaga, K. 1972. *Introduction to Statistical Pattern Recognition.* Academic Press.

Gauch, J. M. 1999. Image Segmentation and Analysis via Multiscale Gradient Watershed Hierarchies. *Image Processing*, **8**(1), 69–79.

Geurts, P., Ernst, D., & Wehenkel, L. 2006. Extremely Randomized Trees. *Machine Learning*, **36**(1), 3–42.

Gould, S., Rodgers, J., Cohen, D., Elidan, G., & Koller, D. 2008. Multi-Class Segmentation with Relative Location Prior. *IJCV*, **80**(3), 300–316.

Gu, C., Lim, J. J., Arbeláez, P., & Malik, J. 2009. Recognition using Regions. *In: CVPR.*

Guigues, L., Le Men, H., & Cocquerez, J.-P. 2003. The Hierarchy of the Cocoons of a Graph and its Application to Image Segmentation. *Pattern Rec. Lett.*, **24**(8), 1059–1066.

Guyon, I., & Elisseeff, A. 2003. An Introduction to Variable and Feature Selection. *JMLR*, **3**, 1157–1182.

Haar, A. 1910. Zur Theorie der orthogonalen Funktionssysteme. *Mathematische Annalen*, **LXIX**, 331–371.

Herms, K. 2007. *Exploration des Skalenraumes bezüglich der Gebäudeextraktion in terrestrischen Farbbildern.* M.Phil. thesis, Universität Bonn.

Hernández, J., & Marcotegui, B. 2009. Morphological Segmentation of Building Facade Images. *In: ICIP 2009.*

Hoare, C. 1971. Proof of a Program: FIND. *Comm. ACM*, **14**(1), 39–45.

Hohmann, B., Krispel, U., Havemann, S., & Fellner, D. 2009. Cityfit: High-Quality Urban Reconstruction by Fitting Shape Grammars to Images and Derived Textured Point Clouds. *In: Proceedings of the 3rd ISPRS International Workshop 3D-ARCH 2009.*

Hoiem, D., Efros, A. A., & Hebert, M. 2008. Closing the Loop on Scene Interpretation. *In: CVPR.*

Holmes, G., Pfahringer, B., Kirkby, R., Frank, E., & Hall, M. 2002. Multiclass Alternating Decision Trees. *Pages 161–172 of: ECML'02.* LNCS 2430. Springer.

Itti, L., Koch, C., & Niebur, E. 1998. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *PAMI*, **20**(11), 1254–1259.

Jahangiri, M. 2010. *An Attention Model and its Application in Man-Made Scene Interpretation.* Ph.D. thesis, Imperial College London.

Jahangiri, M., & Petrou, M. 2009. An Attention Model for Extracting Regions that Merit Identification. *In: ICIP'09.*

Jin, Y., & Geman, S. 2006. Context and Hierarchy in a Probabilistic Image Model. *Pages II: 2145–2152 of: CVPR.*

Kluckner, S. 2011. *Semantic Interpretation of Digital Aerial Images Utilizing Redundancy, Appearance and 3D Information.* Ph.D. thesis, TU Graz.

Koenderink, J. J. 1984. The Structure of Images. *Biological Cybernetics*, **50**(5), 363–370.

Korč, F., & Förstner, W. 2008. Interpreting Terrestrial Images of Urban Scenes using Discriminative Random Fields. *Pages 291–296 of: 21st ISPRS Congress.* IAPRS 37 (B3a).

Korč, F., & Förstner, W. 2009. *eTRIMS Image Database for Interpreting Images of Man-Made Scenes.* Tech. rept. TR-IGG-P-2009-01. IGG University of Bonn.

Korč, F., & Schneider, D. 2007. *Annotation Tool.* Tech. rept. TR-IGG-P-2007-01. IGG University of Bonn.

Kropatsch, W.G. 1995. Building Irregular Pyramids by Dual-Graph Contraction. *Vision Image and Signal Processing*, **142**(6), 366–374.

Kulschewski, K. 1999. *Modellierung von Unsicherheiten in dynamischen Bayes-Netzen zur qualitativen Gebäudeerkennung.* Ph.D. thesis, Universiät Bonn.

Kumar, S., & Hebert, M. 2003a. Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification. *Pages 1150–1157 of: ICCV.*

Kumar, S., & Hebert, M. 2003b. Man-made Structure Detection in Natural Images using a Causal Multiscale Random Field. *Pages I: 119–126 of: CVPR.*

Lafferty, J., McCallum, A., & Pereira, F. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Pages 282–289 of: ICML.*

Laws, K. I. 1980. Rapid Texture Identification. *Pages 376–380 of: Image Processing for Missile Guidance.* SPIE 238.

Lee, S. C., & Nevatia, R. 2004. Extraction and Integration of Window in a 3D Building Model from Ground View Images. *Pages II: 113–120 of: CVPR.*

Lepetit, V., Lagger, P., & Fua, P. 2005. Randomized Trees for Real-Time Keypoint Recognition. *Pages II: 775–781 of: CVPR.*

Lifschitz, I. 2005. *Image Interpretation using Bottom-up Top-down Cycle on Fragment Trees.* M.Phil. thesis, Weizmann Institute of Science.

Lim, J. J., Arbeláez, P., Gu, C., & Malik, J. 2009. Context by Region Ancestry. *Pages 1978–1985 of: ICCV.*

Lin, Y.-C., Tsai, Y.-P., Hung, Y.-P., & Shih, Z.-C. 2006. Comparison between Immersion-based and Toboggan-based Watershed Image Segmentation. *Image Processing*, **15**(3), 632–640.

Lindeberg, T. 1994. *Scale Space Theory in Computer Vision.* Kluwer Academic.

Lowe, D. G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, **60**(2), 91–110.

Marfil, R., Bandera, A., & Sandoval, F. 2007. Perception-based Image Segmentation using the Bounded Irregular Pyramid. *Pages 244–253 of: DAGM.* LNCS 4713.

Matas, J., Chum, O., Urban, M., & Pajdla, T. 2002. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. *Pages I: 384–393 of: BMVC.*

Maxwell, J. C. 1870. On Hills and Dales. *Philiosophical Magazine and Journal of Science*, **40**(4), 421–427.

Meer, P. 1989. Stochastic Image Pyramids. *CVGIP*, **45**(3), 269–294.

Meine, H., & Köthe, U. 2006. A New Sub-pixel Map for Image Analysis. *Pages 116–130 of: IWCIA'06.* LNCS 4040.

Meixner, P., & Leberl, F. 2010. Vertical or Oblique Imagery for Semantic Building Interpretation. *Pages 247–256 of: Publikationen der DGPF: 100 Jahre ISPRS - 100 Jahre internationale Zusammenarbeit.*

Mladenić, D. 2006. Feature Selection for Dimensionality Reduction. *Pages 84–102 of: SLSFS'06.* LNCS 3940.

Müller, P., Wonka, P., Haegler, S., Ulmer, A., & Van Gool, L. 2006. Procedural Modeling of Buildings. *ACM Transactions on Graphics*, **25**(3), 614–623.

Najman, L., & Couprie, M. 2003. Watershed Algorithms and Contrast Preservation. *In: DGCI'03.* LNCS 2886.

Ngom, P., & Emilion, R. 2008. Evaluating Fit using Hellinger Discrimination and Dirichlet Process Prior. *Journal des Sciences et Technologies*, **6**(1), 15–28.

Olsen, O. F. 1996 (September). *Multi-Scale Segmentation of Grey-Scale Images.* M.Phil. thesis, University of Copenhagen.

Ommer, B. 2007. *Learning the Compositional Nature of Objects for Visual Recognition.* Ph.D. thesis, ETH Zürich.

Ommer, B., & Buhmann, J. 2010. Learning the Compositional Nature of Visual Object Categories for Recognition. *PAMI*, **32**(3), 501–516.

Pan, H.-P. 1994. Two-level Global Optimization for Image Segmentation. *P& RS*, **49**(2), 21–32.

Pantofaru, C., Schmid, C., & Hebert, M. 2008. Object Recognition by Integrating Multiple Image Segmentations. *In: ECCV.*

Papoulis, A., & Pillai, S. U. 2002. *Probability, Random Variables and Stochastic Processes.* McGraw-Hill.

Pearl, J. 1997. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* 4th revised and updated ed. edn. Morgan Kaufman.

Perona, P., & Malik, J. 1990. Scale-Space and Edge Detection using Anisotropic Diffussion. *PAMI*, **12**(7), 629–639.

Petrou, M., & Bosdogianni, P. 1999. *Image Processing: The Fundamentals.* Wiley.

Pfahringer, B., Holmes, G., & Kirkby, R. 2001. Optimizing the Induction of Alternating Decision Trees. *Pages 477–487 of: PACKDDM'01.* LNCS 2035.

Plath, N., Toussaint, M., & Nakajima, S. 2009. Multi-class Image Segmentation using Conditional Random Fields and Global Classification. *Pages 817–824 of: 26th ICML.*

Rätsch, G. 2003. Robust Multi-Class Boosting. *Pages 997–1000 of: Proc. EuroSpeech.*

Rätsch, G., & Warmuth, M.K. 2005. Efficient Margin Maximizing with Boosting. *JMLR*, **6**, 2131–2152.

Rätsch, G., Onoda, T., & Müller, K.-R. 2001. Soft Margins for AdaBoost. *Machine Learning*, **43**(3), 287–320.

Reznik, S. 2009. *Aussehensbasierte generative hierarchische Interpretation von Fassaden in terrestrischen Bildsequenzen.* Ph.D. thesis, Universität der Bundeswehr München.

Ripperda, N. 2008. Grammar Based Facade Reconstruction using rjMCMC. *PFG*, **2008**(2), 83–92.

Ripperda, N. 2009. *Rekonstruktion von Fassadenstrukturen mittels formaler Grammatiken und Reversible Jump Markov Chain Monte Carlo Sampling.* Ph.D. thesis, Leibniz Universität Hannover.

Ripperda, N., & Brenner, C. 2009. Evaluation of Structure Recognition using Labelled Facade Images. *Pages 532–541 of: DAGM.* LNCS 5748.

Roth, V., & Steinhage, V. 1999. Nonlinear Discriminant Analysis using Kernel Functions. *Pages 568–574 of: NIPS'99.*

Russell, S., & Norvig, P. 1995. *Artificial Intelligence: A Modern Approach.* 1st ed. edn. Pearson Education.

Schapire, R. E. 1990. The Strength of Weak Learnability. *Machine Learning*, **5**(2), 197–227.

Schapire, R. E., & Singer, Y. 1999. Improved Boosting Algorithms using Confidence-Rated Predictions. *Machine Learning*, **37**(3), 297–336.

Schnitzspan, P., Fritz, M., & Schiele, B. 2008. Hierarchical Support Vector Random Fields: Joint Training to Combine Local and Global Features. *Pages 527–540 of: ECCV.*

Schuster, H.-F. 2002. *Bildsegmentierung mit stochastischen Bildpyramiden.* M.Phil. thesis, University of Bonn.

Schuster, H.-F. 2005. *Detection of Man-made-Objects based on Spatial Aggregations.* Tech. rept. TB-ipb-05-1. Institute for Photogrammetry, Bonn University.

Shotton, J., Winn, J., Rother, C., & Criminisi, A. 2006. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. *Pages 1–15 of: ECCV.* LNCS 3951.

Shotton, J., Johnson, M., & Cipolla, R. 2008. Semantic Texton Forests for Image Categorization and Segmentation. *In: CVPR.*

Sirmacek, B., & Ünsalan, C. 2008. Building Detection from Aerial Imagery using Invariant Color Features and Shadow Information. *In: International Symposium on Computer and Information Sciences.*

Steger, C. 1999. Subpixel-Precise Extraction of Watersheds. *Pages II: 884–890 of: ICCV.*

Steinmetz, S., Paulus, D., & Hans, W. 2007. Schattenentfernung unter Verwendung des Retinex-Algorithmus. *Pages 93–104 of: 13. Workshop Farbbildverarbeitung.*

Teboul, O., Simon, L., Koutsourakis, P., & Paragios, N. 2010. Segmentation of Building Facades using Procedural Shape Priors. *In: CVPR.*

Terzić, K., Hotz, L., & Šochman, J. 2010. Interpreting Structures in Man-Made Scenes: Combining Low-Level and High-Level Structure Sources . *In: International Conference on Agents and Artificial Intelligence.*

Treisman, A. 1986. Features and Objects in Visual Processing. *Scientific American*, **225**, 114–125.

Tu, Z., Chen, X., Yuille, A. L., & Zhu, S.-C. 2005. Image Parsing: Unifying Segmentation, Detection, and Recognition. *IJCV*, **63**(2), 113–140.

Verbeek, J., & Triggs, B. 2007. Region Classification with Markov Field Aspect Models. *Pages 1–8 of: CVPR.*

Vincent, L., & Soille, P. 1991. Watersheds in Digital Spaces: An Efficient Algorithm based on Immersion Simulations. *PAMI*, **13**(6), 583–598.

Viola, P., & Jones, M. 2001a. Rapid Object Detection using a Boosted Cascade of Simple Features. *Pages 511–518 of: CVPR.*

Viola, P., & Jones, M. 2001b. Robust Real-Time Object Detection. *In: 2nd Internat. Workshop on Statistical and Computational Theories of Vision.*

Wahl, R., Schnabel, R., & Klein, R. 2008. From Detailed Digital Surface Models to City Models using Constrainted Simplification. *PFG*, **2008**(3), 207–215.

Warmuth, M. K., Glocer, K., & Rätsch, G. 2008. Boosting Algorithms for Maximizing the Soft Margin. *In: NIPS.*

Wenzel, S., & Förstner, W. 2008. Semi-Supervised Incremental Learning of Hierarchical Appearance Models. *Pages 399–404 of: 21st ISPRS Congress.* IAPRS 37 (B3b-2).

Winter, S. 1995. Topological Relations between Discrete Regions. *Pages 310–328 of: 4th Symposium on Large Spatial Databases.* LNCS 951.

Witkin, A. 1983. Scale-Space Filtering. *Pages 1019–1022 of: IJCAI'83.*

Xu, M., Petrou, M., & Jahangiri, M. 2010. Component Identification in the 3D Model of a Building. *Pages 3061–3064 of: ICPR 2010.*

Yang, M. Y., Förstner, W., & Drauschke, M. 2010. Hierarchical Conditional Random Field for Multi-Class Image Classification. *Pages 464–469 of: International Conference on Computer Vision Theory and Applications (VISAPP).*

Zhu, J., Rosset, S., Zou, H., & Hastie, T. 2006. *Multi-Class AdaBoost.* Tech. rept. Dep. of Statistics, University of Michigan.