DEUTSCHE GEODÄTISCHE KOMMISSION
bei der Bayerischen Akademie der Wissenschaften

Reihe C | Dissertationen | Heft Nr. 586

Karl-Heinrich Anders

Parameterfreies hierarchisches Graph-Clustering Verfahren zur Interpretation raumbezogener Daten

München 2007

Verlag der Bayerischen Akademie der Wissenschaften
in Kommission beim Verlag C. H. Beck

ISSN 0065-5325 | ISBN 3 7696 5025 5

DEUTSCHE GEODÄTISCHE KOMMISSION
bei der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 586

Parameterfreies hierarchisches Graph-Clustering Verfahren zur Interpretation raumbezogener Daten

Von der Fakultät für Luft- und Raumfahrttechnik und Geodäsie
der Universität Stuttgart
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Abhandlung

vorgelegt von

Dipl.-Ing. Karl-Heinrich Anders

München 2007

Verlag der Bayerischen Akademie der Wissenschaften
in Kommission beim Verlag C. H. Beck

ISSN 0065-5325 ISBN 3 7696 5025 5

Adresse der Deutschen Geodätischen Kommission:

Deutsche Geodätische Kommission

Alfons-Goppel-Straße 11 ● D – 80 539 München

Telefon (089) 23 031 -1113 ● Telefax (089) 23 031 -1283/ -1100

E-mail hornik@dgfi.badw.de ● http://dgk.badw.de

Hauptberichter: Prof. Dr.-Ing. Dieter Fritsch

Mitberichter: Prof. Dr.-Ing. Monika Sester

Prof. Dr.rer.nat. Ralf Reulke

Tag der mündlichen Prüfung: 3. Dezember 2003

ISSN 0065-5325 ISBN 3 7696 5025 5

Inhaltsverzeichnis

Zusammenfassung

Die Notwendigkeit der automatischen Interpretation und Analyse von räumlichen Daten wird heutzutage immer wichtiger, da eine stetige Zunahme der digitalen räumlichen Daten zu verzeichnen ist. Dies betrifft auf der einen Seite Rasterdaten wie auch auf der anderen Seite Vektordaten, welche überwiegend auf unterschiedlichen Landschaftsmodellen basieren. Differenzen zwischen diesen Landschaftsmodellen bestehen u.a. in den Objektarten, dem Grad der Generalisierung oder der geometrischen Genauigkeit der gespeicherten Landschaftsobjekte. Die interaktive Prozessierung und Analyse von großen Datenbeständen ist sehr zeitaufwendig und teuer. Speziell die manuelle Analyse räumlicher Daten zum Zwecke der Datenrevision wird in Zukunft das Limit der technischen Umsetzbarkeit erreichen, da moderne Anforderungen an die Laufendhaltung der Daten zu immer kürzeren Aktualisierungszyklen führen.

Die automatische Interpretation digitaler Landschaftsmodelle setzt die Integration von Methoden des räumlichen Data Mining bzw. Knowledge Discovery in raumbezogenen Daten innerhalb von Geographischen Informationssystemen (GIS) voraus. Hierauf wird sich die vorliegende Arbeit konzentrieren. Grundsätzlich kann man die automatische Interpretation von digitalen Landschaftsmodellen (DLM) in drei Kategorien unterscheiden: die Interpretation basierend auf einem spezifischen Modell des DLM, die Interpretation basierend auf einem generischen Modell der Basiselemente des DLM sowie die unüberwachte Interpretation des DLM.

Zunächst beschreiben wir einen Ansatz zur Generierung von 3D-Gebäuden, welche als Hypothese aus Katasterkarten abgleitet werden. Diese Vorgehensweise stellt ein Beispiel für die DLM-Interpretation auf der Grundlage eines spezifischen Modells dar und kann zur schnellen Generierung von groben 3D-Stadtmodellen oder als Vorabinformation zur bildgestützten 3D-Gebäuderekonstruktion verwendet werden. Des weiteren stellen wir detailliert einen Ansatz zur Ableitung von ATKIS-Daten aus ALK-Daten vor, welcher ein Beispiel für die DLM-Interpretation basierend auf einem generischen Modell der DLM-Basiselemente darstellt und zur automatischen Laufendhaltung der Daten dient. Beide Ansätze führen direkt zum grundsätzlichen Problem der Gruppierung von räumlichen Objekten, welches generell unter dem Begriff des Clusterns zusammengefasst wird. Man unterscheidet zwei Arten von Clusterverfahren: überwachte und unüberwachte Methoden. Unüberwachte Cluster- oder Lernverfahren können für den dritten genannten Fall der DLM-Interpretation verwendet werden und sind gut geeignet für die Modellgeneralisierung und die kartographische Generalisierung von DLM-Daten, falls die Methoden in der Lage sind, Cluster mit beliebiger Form zu erkennen. Die bisher existierenden Verfahren benötigen jedoch zumeist verschiedenste Kenntnisse als Voraussetzung, wie z.B. die Verteilungsfunktion der Daten oder Schrankenwerte für Ähnlichkeitsmessungen bzw. Abbruchkriterien. Zudem finden viele Clusterverfahren nur Gruppierungen mit konvexer Form und erkennen keine Löcher (z.B. Maximum-Likelihood-Methoden).

Der Hauptteil dieser Arbeit widmet sich einem neu entwickelten, unüberwachten Clusterverfahren zur automatischen Interpretation von raumbezogenen Daten. Das Verfahren heißt Hierarchisches Parameterfreies Graph-CLustering (HPGCL) und dient zur Erkennung von Clustern beliebiger Form. Es benötigt weder Parameter wie z.B. Schrankenwerte noch Annahmen über die Verteilung der Daten oder die Anzahl der Cluster. Die Neuartigkeit des HPGCL-Algorithmus besteht auf der einen Seite in der Anwendung der Hierarchie von Nachbarschaftsgraphen zur Definition der Nachbarschaft eines Einzelobjekts oder eines Objektclusters in allgemeiner Art und Weise, sowie auf der anderen Seite in der Definition eines Entscheidungskriteriums zur Ähnlichkeitsbestimmung von Clustern, welches medianbasiert ist und ohne Angabe von Schwellwerten auskommt. Der Nächste-Nachbar-Graph, der Minimal Spannende Baum, der Relative Nachbarschaftsgraph, der Gabriel-Graph und die Delaunay-Triangulation kommen im HPGCL-Algorithmus zum Einsatz. Es wird aufgezeigt, dass die hierarchische Beziehung dieser Nachbarschaftsgraphen in einem natürlichen Generalisierungsprozess im Sinne einer grob-zu-fein-Segmentierung eines Datensatzes genutzt werden kann. Als weiterer Aspekt des HPGCL-Algorithmus kann die Tatsache genannt werden, dass im allgemeinen eine begrenzte Anzahl von Clustern größer eins gefunden wird. Im Gegensatz dazu benötigen andere hierarchische Clusterverfahren generell die Minimalanzahl der zu findenden Cluster als Parameter, da ohne Abbruchkriterium sonst alle Objekte des Datensatzes in einem einzigen großen Cluster vereinigt werden. Die Arbeit untersucht detailliert den Einfluss eines einzelnen Nachbarschaftsgraphen in der Hierarchie auf das Ergebnis des Clusterings, und es wird die Verwendbarkeit des HPGCL-Algorithmus auf der Grundlage von verschiedenen Datensatztypen evaluiert. Anhand zweier Datensätze werden die Ergebnisse des HPGCL-Verfahrens mit den Resultaten eines durch Testpersonen durchgeführten manuellen Clusterings verglichen.

Abstract

Nowadays, the necessity of automatic interpretation and analysis of spatial data is getting more and more important, because the amount of digital spatial data continuously increases. On the one hand, there are raster data sets, on the other hand vector data that are predominantly based on different landscape models. Differences between these landscape models are, e.g., the object type, the degree of generalization or the geometric accuracy of the captured landscape objects. The pure interactive processing and analysis of large spatial databases is very time-consuming and expensive. Especially the manual analysis of spatial data for the purpose of data revision will reach the limit of technical feasibility in the near future, because modern requirements on the up-to-dateness of data lead to ever shorter update cycles.

The automatic interpretation of digital landscape models needs the integration of methods of the field of spatial data mining or knowledge discovery in spatial databases into geographical information systems (GIS), which is the focus of this thesis. In general, the automatic interpretation of a digital landscape model (DLM) can be divided into the interpretation based on a specific model of the DLM, the interpretation based on a generic model of the basic elements of the DLM and the unsupervised interpretation of the DLM.

First, an approach for the generation of 3D building hypotheses from a cadastral map is described which is an example for the DLM interpretation based on a specific model. This approach can be used for the fast generation of approximate 3D city models or as pre-information for an image based 3D building reconstruction. Secondly, an approach for the automatic derivation of ATKIS data from ALK data will be described in detail, which is an example for the DLM interpretation based on a generic model of the basic DLM elements for the purpose of automatic data revision. Both approaches lead directly to the basic problem of grouping spatial objects, which can be seen as a general clustering problem. Clustering methods can be divided into supervised and unsupervised methods. Unsupervised clustering or learning methods can be used for the third case of DLM interpretation. Especially unsupervised clustering methods are well suited for the model generalization and the cartographic generalization of DLM data if these methods can recognize clusters of arbitrary shape. There are a lot of different clustering approaches, but most of them need certain prerequisites, like the distribution function of the data or thresholds for similarity tests and terminating conditions. In many cases, clustering methods can only find clusters with a convex shape and without holes (e.g. maximum-likelihood).

The main contribution of this thesis is a new unsupervised clustering method called Hierarchical Parameter-free Graph CLustering (HPGCL) for the automatic interpretation of spatial data. The HPGCL algorithm can find clusters of arbitrary shape and needs neither parameters like thresholds nor an assumption about the distribution of the data or number of clusters. The novelty of the HPGCL algorithm lies on the one hand in the application of the hierarchy of neighbourhood graphs (also called proximity graphs) to define the neighbourhood of a single object and object clusters in a natural and common way and on the other hand in the definition of a median based, threshold free decision criteria for the similarity of clusters. In the HPGCL algorithm the Nearest-Neighbour-Graph, the Minimum-Spanning-Tree, the Relative-Neighbourhood-Graph, the Gabriel-Graph and the Delaunay-Triangulation are used. It will be shown that the hierarchical relationship of these proximity graphs can be used for a natural generalization process in the sense of a coarse-to-fine segmentation of a data set. One additional feature of the HPGCL algorithm is that in general a limiting number of clusters greater than one will be found. In contrast, general hierarchical cluster algorithms require the minimal number of clusters as a parameter, otherwise they will always group all objects of a data set in one big cluster. The influence of the single proximity graphs of the hierarchy on the clustering result is investigated in detail. The usability of the HPGCL algorithm is evaluated on different types of data sets and for two data sets the results of the HPGCL algorithm are compared with manual clustering results.

Kapitel 1

Einleitung

1.1 Motivation und Aufgabenstellung

Die Notwendigkeit der automatisierten Interpretation und Analyse von raumbezogenen Daten wird heutzutage immer deutlicher, da die große Menge an raumbezogenen Daten, die in digitaler Form vorliegen, stetig ansteigt. All diese raumbezogenen Daten basieren überwiegend auf verschiedenen Modellen der Landschaft. Unterschiede in diesen Modellen betreffen z. B. die Objektart, den Grad an Generalisierung oder die geometrische Genauigkeit der erfassten Landschaftsobjekte. Eine rein interaktive Bearbeitung und Analyse großer raumbezogener Datenbanken ist extrem zeit- und kostenintensiv. Besonders die operationelle Analyse raumbezogener Daten zu Fortführungszwecken wird in naher Zukunft an ihre Grenzen der Realisierbarkeit stoßen.

Diese Problematik erfordert, Methoden des sogenannten *Spatial Data Mining* oder *Knowledge Discovery in Spatial Databases (KDSD)* in Geo-Informationssystemen zu integrieren. Unter dem Begriff Knowledge Discovery in Spatial Databases versteht man die Ableitung markanter, impliziter und vorher unbekannter Informationen aus großen räumlichen Datenbanken. Das Aufgabenfeld von KDSD integriert die Gebiete des Maschinellen Lernen, Datenbanksysteme, Datenvisualisierung, Statistik, Informationstheorie und algorithmischer Geometrie.

Die automatische Interpretation von digitalen Landschaftsmodellen (DLM) kann man in drei Fälle einteilen :

1. Interpretation basierend auf einem spezifischen Modell des DLM.
2. Interpretation basierend auf einem generischen Modell der DLM-Grundelemente.
3. Interpretation ohne Vorinformationen über das DLM.

Im Fall 1 nutzt man zur Interpretation spezifische Informationen über das DLM, um neue Informationen abzuleiten. Mit einem spezifischen Modell ist hier die Verwendung von Instanzen expliziter Objektmodelle gemeint, wie z.B. Polygone, von denen man weiß, dass sie Gebäude oder Flurstücke darstellen.

Im Fall 2 besitzt man spezifisches Wissen über die Grundelemente des DLM, wie z.B. Text, Zahl, Punkt oder Linie. Dieses spezifische Wissen wird dann um ein explizites Regelwerk erweitert, das angibt, wie diese Grundelemente in einem bestimmten Kontext zu verstehen sind. Ein Beispiel hierfür ist die Strukturierung von sogenannten „Spaghetti-Daten“ (unstrukturierte Menge von digitalen graphischen Daten). Das verwendete Regelwerk muss jedoch manuell vorgegeben werden.

Der Fall 3 unterscheidet sich vom Fall 2 darin, dass kein explizites Regelwerk vorgegeben ist. In diesem Fall gibt man ein „Hypothesenmodell“ (Wahrscheinlichkeitsmodell) vor, das beschreibt, welche Grundelemente in einem bestimmten Kontext als ähnlich oder zusammengehörig betrachtet werden können und das automatische Verfahren soll dann nach „wahrscheinlichsten“ (signifikantesten) Gruppierungen suchen und möglichst eine kompakte Charakterisierung (Beschreibung) dieser Gruppen liefern. Die Verfahren des Spatial Data Mining gehören deshalb zum Fall 3.

Für alle drei Fälle werden wir im folgenden Beispiele geben, wobei die Konzentration auf den 3. Fall erfolgt.

Die manuelle Erfassung von 3D-Stadtmodellen für städtische Gebäudeinformationssysteme, Unternehmen der Telekommunikation oder für Touristikzwecke (virtuelle Stadtmodelle von Urlaubsorten im Internet) ist sehr zeit- und kostenintensiv. Eine vollständig automatische und flächendeckende 3D-Rekonstruktion von Gebäuden aus Luftbildern oder Laserdaten ist bis heute nicht möglich (Förstner 1999). Durch zusätzliche Informationen aus einem Geoinformationssystem (GIS) lässt sich die Problematik der automatischen Rekonstruktion erheblich einschränken. In (Brenner 2000) wird ein vollständig automatisches Verfahren zur flächendeckenden 3D-Rekonstruktion von Gebäuden aus Laserdaten vorgestellt, das als einzige Zusatzinformation 2D-Gebäudegrundrisse benötigt. Die Bereitstellung solch geeigneter Zusatzinformationen gehört zum Fall 1, der in dieser Arbeit anhand der Erzeugung von 3D-Gebäudehypothesen aus 2D-Gebäudedaten untersucht wird. Die hierfür verwendeten raumbezogenen Daten stammten aus der *automatisierten Liegenschaftskarte (ALK)* der öffentlichen Vermessungsverwaltungen[1], die Informationen über die Lage und Form von Gebäudegrundrissen, sowie die Gebäudenutzung enthält. Eine weitere Fragestellung ergab sich dabei zum Nutzen solcher grober 3D-Gebäudehypothesen als Hintergrundvisualisierung in großen 3D-Stadtmodellen. Den 2. Fall werden wir anhand der automatischen Ableitung von ATKIS-Daten aus ALK-Daten beschreiben.

Die Untersuchungen der Fälle 1 und 2 führten beide zur grundlegenden Problematik der Gruppierung raumbezogener Objekte das als *Clustering-Problem* aufgefasst werden kann und somit zum Gebiet des Spatial Data Mining gehört. Auf dem Gebiet des Clustering existieren viele Ansätze von denen die meisten direkt oder indirekt Voraussetzungen über die Verteilung der Daten treffen, Schwellwerte voraussetzen und häufig auch nur spezielle Clusterformen (konvexe Formen) bestimmen können. Ziel war es nun, ein vollständig parameterfreies Verfahren zum Clustering räumlicher Objekte zu ermitteln, das es erlaubt, Objektgruppen beliebiger Form zu bilden. Die so gefundenen Gruppen können dann bei der Modellgeneralisierung oder der kartographischen Generalisierung (speziell Typisierung) entsprechend weiterverarbeitet werden.

1.2 Aufbau der Arbeit

Zur Motivation für diese Arbeit und zur Einführung in die raumbezogene Clusteranalyse und ihre Anwendungsmöglichkeiten beschreiben wir zuerst im Kapitel 2 zwei Beispiele der automatischen Interpretation von GIS-Daten. Im ersten Beispiel zeigen wir die Verwendung von GIS-Daten als Vorinformation zur Bildinterpretation im Falle der 3D-Gebäuderekonstruktion aus Luftbildern oder Laserdaten. Das zweite Beispiel beschreibt, anhand der automatischen Ableitung von ATKIS-Daten aus ALK-Daten, die automatische *Modellgeneralisierung*, d.h. die Ableitung kleinmaßstäbiger geographischer Daten aus großmaßstäbigen Daten. In beiden Beispielen stoßen wir auf das Problem der Gruppierung von topologisch unstrukturierten räumlichen Objekten.

In Kapitel 3 gehen wir kurz auf die wesentlichen Aspekte des sogenannten *Data Mining* ein, um eine Einordnung der raumbezogenen Clusteranalyse innerhalb der Aufgaben (Methoden) der automatsierten Dateninterpretation zu geben. Eine detaillierte Beschreibung der Aufgaben und Methoden der Clusteranalyse geben wir dann im Kapitel 4, um unseren in dieser Arbeit entwickelten Ansatz zur Gruppierung raumbezogener Daten besser einordnen zu können.

In jedem Clusterverfahren spielt der Vergleich von Daten (Objekten), d.h. die Definition und Berechnung der *Ähnlichkeit* oder des Abstands von Objekten, *die* wesentliche Rolle. In Kapitel 5 gehen wir deshalb ausführlich auf die Problematik von Ähnlichkeits- oder Distanzmaßen ein und geben ein Distanzmaß für Konzepte einer Konzepthierarchie an und geben eine Begründung für unsere Wahl der Nachbarschaftsgraphen als Repräsentant der Ähnlichkeit zwischen Objekten. Da Nachbarschaftsgraphen in unserem Verfahren die wesentliche Rolle spielen, ist eine ausführliche Beschreibung von Nachbarschaftsgraphen und der von uns verwendeten Graphen im Kapitel 6 gegeben.

Unser hierarchisches parameterfreies Graph-Clustering (HPGCL) beschreiben wir dann ausführlich in Kapitel 7 und geben dabei unsere Definition eines Clusters, sowie ein medianbasiertes, schwellwertfreies Entscheidungskriterium zur Vereinigung ähnlicher Cluster. In Kapitel 8 wenden wir dann unser Verfahren auf unterschiedlichen Daten an und beschreiben die erzielten Ergebnisse. Für die durchgeführten Tests standen künstlich erzeugte Daten, Daten aus dem Gebäudeinformationssystem des Stadtvermessungsamts Stuttgart, sowie Laserdaten und Bilddaten zur Verfügung, welche das breite Spektrum des Verfahrens demonstrieren.

Im letzten Kapitel 9 werden dann die erzielten Ergebnisse diskutiert und ein Ausblick auf mögliche zukünftige Arbeiten angezeigt.

[1] In Bayern DFK (Digitale Flurkarte) genannt.

1.3 Abgrenzung zu anderen Arbeiten

Unsere Arbeiten zur Ableitung von 3D-Gebäudehypothesen aus GIS-Daten zur Rekonstruktion von Gebäuden aus Luftbildern und Laser-Daten (Haala & Anders 1996, Haala & Anders 1997) waren unseres Erachtens die ersten Arbeiten zu dieser Problematik. Die prinzipielle Möglichkeit der automatischen Fortführung von ATKIS Basis-DLM aus ALK-Daten wurde in den Arbeiten (Anders & Sester 1997, Anders, Sester & Fritsch 1997, Anders 1997, Sester, Anders & Walter 1998), soweit uns bekannt, erstmals aufgezeigt.

Der in dieser Arbeit vorgestellte Ansatz zur Gruppierung raumbezogener Daten kann als hierarchisches, agglomeratives, dichtebasiertes Clusterverfahren eingeordnet werden. Ester, Kriegel, Sander & Xu (1996) beschreiben ebenfalls ein dichtebasiertes Clustering-Verfahren, das jedoch nicht auf Nachbarschaftsgraphen aufbaut, wie unser Verfahren. Die Autoren definieren, basierend auf einer ϵ-Umgebung, die Begriffe *density-reachable* und *density-connected*, um benachbarte (ähnliche) Objekte und Cluster zu charakterisieren. Das ϵ muss dabei vorgegeben werden oder kann, unter der Annahme einer räumlichen Verteilung, geschätzt werden.

Wie oben herausgestellt, unterscheiden sich graphbasierte Verfahren von unserem Verfahren im wesentlichen durch die Art der verwendeten Graphen, die Verwendung von Schwellwerten und der Definition der Clusterdichte, soweit diese Verfahren die Dichte als Maß für die Ähnlichkeit verwenden. Unseres Wissens nach sind (Anders, Sester & Fritsch 1999, Anders & Sester 2000, Anders 2001) auch die ersten Veröffentlichungen zur expliziten Verwendung der *Hierarchie der Nachbarschaftsgraphen* zum parameterfreien Clustering räumlicher Objekte. Das graphbasierte Verfahren von Zahn (1971) verwendet nur den minimal spannenden Baum und einen Schwellwert als Entscheidungskriterium zur Entfernung „ungeeigneter“ Graphkanten. Urquhart (1982) untersuchte den Gabriel Graph (siehe Abschnitt 6.2.5) auf seine Eignung zum Clustering. Das dichtebasierte Verfahren von Bajcsy & Ahuja (1998) bestimmt, im Gegensatz zu uns, den *vollständigen Graphen* (siehe Abschnitt 6.1.2). In den Arbeiten von Jarvis & Patrick (1973) und Eppstein (1998) wird der *k-Nächster-Nachbargraph* verwendet, wodurch k als Parameter ins Spiel kommt; in (Jarvis & Patrick 1973) wird noch ein weiterer Schwellwert für die Gruppierung benachbarter Objekte verwendet. Estivill-Castro & Lee (2002) verwenden die *Delaunay-Triangulation* (Abschnitt 6.2.7) und ein spezielles Kriterium zur Entfernung „ungeeigneter“ Graphkanten, das sie als *Short-Long-Kriterium* bezeichnen. In der Arbeit von van Schröder (2001) wird die *gleichmässige Baumzerlegung* verwendet, die, wie der Name schon sagt, einen Baum (siehe Def. 6.1.10) als Graph voraussetzt. van Dongen (2000) beschreibt ein stochastisches Graph-Clustering-Verfahren, das auf einem *Markov-Prozess* aufbaut.

Damit grenzt sich unser Verfahren in folgenden Punkten von bestehenden Arbeiten ab:

- Das gesamte Verfahren benötigt keine Vorgabe oder Schätzung von Parametern und es sind keine Annahmen über die zugrundeliegende Verteilung der Merkmale der zu gruppierenden Objekte notwendig.
- Die Verwendung mehrerer Arten von Nachbarschaftsgraphen anstelle eines einzelnen Graphen und ihre explizite Verknüpfung durch die Teilmengenbeziehung ihrer Kantenmengen.
- Die Definition eines Clusters als Kombination äußerer und innerer Kantenmengen ermöglicht eine unscharfe Modellierung der Clusterdichte.
- Die medianbasierte Definition eines Ähnlichkeitsmaßes zum robusten Vergleich von Clustern.

Kapitel 2

Interpretation raumbezogener Daten

2.1 Ableitung von 3D-Gebäudehypothesen

Die Nachfrage nach digitalen 3D-Stadtmodellen von Seiten der Stadtplaner nimmt kontinuierlich zu. Daraus ergab sich die Frage, ob sich vorhandene 2D-Gebäudedaten nutzen lassen, um den Prozeß der 3D-Gebäuderekonstruktion aus Luftbildern oder Laserdistanzdaten zu unterstützen.

Bereits vorhandene Gebäudegrundrisse in einem zweidimensionalen Geo-Informationssystem sind eine zuverlässige Zusatzinformationsquelle für die Erfassung qualifizierter Gebäudehöhenmodelle aus direkt mit Laserscanner (Abb. 2.1) bzw. indirekt durch Stereobildzuordnung (Abb. 2.2) erfassten Digitalen Höhenmodellen (DHM) oder auch direkt aus Stereobildern. Eine automatische Interpretation dieser 3D-Punktwolken ist derzeit noch immer Forschungsgegenstand (Baltsavias, Grün & Van Gool 2001). Die momentan vielversprechensten Ansätze nutzen Vorinformation über die zu erkennenden und zu messenden Objekte aus. Je besser die Vorinformation, desto zuverlässiger sind die Ergebnisse. Ein Beispiel für solche Daten ist das amtliche Liegenschaftskataster (ALK), in dem sowohl Grundrißinformationen als auch die jeweilige Nutzungsart der Gebäude enthalten sind.

Durch die Analyse bzw. Interpretation der vorhandenen Grundrisse können beispielsweise Kontrollparameter für die Segmentierung des DHM bzw. für die anschließende dreidimensionale Rekonstruktion bestimmt werden. Eine wichtige Vorinformation ist der Gebäudegrundriss selbst: Hiermit kann der Bildbereich festgelegt werden, innerhalb dessen sich das Gebäude befindet. Weiterhin lassen sich Größe, Form und Orientierung des Gebäudes ermitteln. Unbekannt ist hingegen die Gebäudehöhe und die Dachform. Diese soll im folgenden mittels Interpretationsverfahren ermittelt werden. Abbildung 2.3 zeigt eine Luftaufnahme des Testgebiets und Abbildung 2.4 den dazu gehörenden Ausschnitt aus der ALK.

2.1.1 Analyse der Gebäudeinformationen

Bei geringen Detailliertheitsanforderungen an die zu erfassende dreidimensionale Form eines Gebäudes kann eine Beschreibung durch einfache Polyeder erfolgen. In diesem Fall kann der Gebäudegrundriß aus dem existierenden GIS ohne weitere Analyse verwendet werden; die benötigte Gebäudehöhe kann dann als die maximale Höhe innerhalb des Grundrisses aus dem DHM bestimmt werden. Für Anwendungen wie Visualisierungen ist häufig ein mittlerer Detailliertheitsgrad ausreichend, für den neben dem Grundriß noch der Dachtyp (z. B. Flach- oder Satteldach) sowie einige wenige Höhenwerte (z. B. First-, Trauf- und Fußbodenhöhe) zu bestimmen sind. Durch die Analyse der ALK-Daten können aus der Form und Lage der Gebäudegrundrisse und den ebenfalls vorhandenen Nutzungsarten Hypothesen (Abb. 2.6, Abb. 2.7) über die dreidimensionale Form gebildet werden. Die unterschiedlichen Hypothesen können anschließend mit Hilfe der DHM- und Bilddaten verifiziert bzw. verworfen werden.

Die Erzeugung von 3D-Gebäudemodellen mit Hilfe der Grundrissinformationen kann natürlich nicht eindeutig gelöst werden. Zusätzlich zu den unbekannten Gebäudehöhen sind die unterschiedlichsten Dachformen möglich, wie z. B. Flach-, Pult-, Sattel- oder Walmdach. Da die Gebäudenutzung einen möglichen Hinweis auf die Dachform und Dachhöhen bietet, nutzen wir die in der ALK enthaltenen Informationen über die Gebäudenutzung,

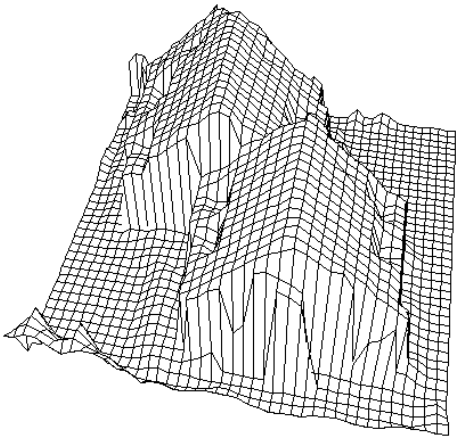


Abbildung 2.1: Ausschnitt aus einem DHM, erfasst mit Laserscanner

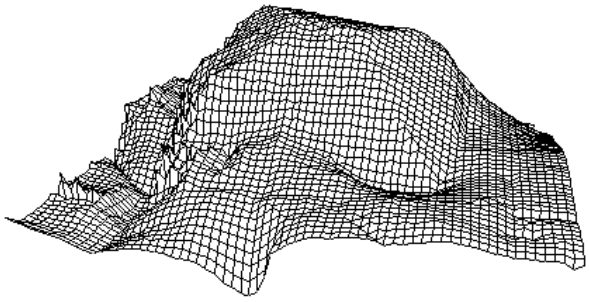


Abbildung 2.2: Ausschnitt aus einem DHM, erfasst durch Stereobildzuordnung

Abbildung 2.3: Bildausschnitt vom Testgebiet

Abbildung 2.4: Ausschnitt aus der verwendeten ALK

um die Gebäude in die sechs Klassen *Wohn-, Industrie-, Bürogebäude, Kirche, Turm* und *Garage* einzuteilen. Jeder dieser Klassen ordnen wir einen Dachtyp und eine Dachhöhe wie folgt zu :

- Garage : Flachdach, Dachhöhe 3m
- Wohn-, Bürogebäude : Satteldach, Traufhöhe 6m, Firsthöhe 9m (das entspricht 2 Geschossen mit je 3m Höhe und ein Dachgeschoß von 3m Höhe)
- Industriegebäude : Flachdach, Dachhöhe 15m (3 Geschosse mit je 5m Höhe)
- Kirche : Flachdach, Dachhöhe 12m
- Turm : Flachdach, Dachhöhe 25m
- Kindergarten : Flachdach, Dachhöhe 5m
- andere : Flachdach, Dachhöhe 7,5m

Diese Werte sind natürlich rein heuristisch, aber sie sollen auch nur eine erste Näherung für das Gebäude liefern. Durch den anschließenden Verifikationsprozeß wird die Hypothese dann bestätigt und verbessert, oder sie wird verworfen und eine neue Hypothese wird gebildet (z.B. andere Firstrichtung oder anderer Dachtyp).

Bevor wir jedoch das 3D-Modell erzeugen, führen wir noch eine Generalisierung des Gebäudegrundrisses durch. Der Dachrand von Flach- und Pultdächern folgt im allgemeinen der Form des Grundrisses. Bei Sattel- oder Walmdächern ist das jedoch meistens nicht der Fall, da kleine Erker oder Unregelmäßigkeiten im Grundriß von der Dachform verdeckt werden. Man kann in den meisten Fällen davon ausgehen, dass die Grundfläche eines solchen Daches rechteckig ist. Als Generalisierung des Grundrisses berechnen wir deshalb ein minimal umschließendes Rechteck, wie es in Abbildung 2.5 dargestellt ist. Die Firstrichtung wird dann durch die längste Seite dieses Rechtecks festgelegt. Basierend auf all den aufgezählten Annahmen erzeugen wir dann ein hypothetisches 3D-Stadtmodell.

In Abbildung 2.6 ist das Ergebnis dargestellt. Ein Vergleich dieses Ergebnisses mit Abbildung 2.3 weist jedoch noch erhebliche Unterschiede auf. So fallen südlich des größeren Industriegebäudes eine Reihe benachbarter Gebäude mit Satteldach auf, die in Realität von einem gemeinsamen Dach bedeckt sind, dessen Firstrichtung senkrecht zu den Grundrissen verläuft. Dies geht darauf zurück, dass wir bisher nur die Grundrisse allein betrachtet haben. Ein Grundriß in der ALK repräsentiert ein betimmtes Besitzverhältnis und angrenzende Grundrisse gehören im allgemeinen zu einem Gebäude und besitzen somit ein gemeinsames Dach. Deshalb bestimmen wir topologisch benachbarte Grundrisse (Grundrisse, die sich ein Geometrieelement vom Typ Linie teilen) und gruppieren diese Grundrisse, falls sie die gleiche Nutzungsart aufweisen, zu einem neuen Grundriss zusammen (Umhüllendes Polygon). Dieses neue Polygon wird dann gegebenenfalls generalisiert und anschließend wird dann ein 3D-Modell erzeugt. Das Ergebnis dieser verbesserten Hypothesenbildung zeigt Abbildung 2.7. Ein nochmaliger Vergleich mit Abbildung 2.3 zeigt, dass sich allein mit diesen einfachen Annahmen ein relativ gutes Stadtmodell erzeugen lässt, was für einfache Visualisierungen sehr gut genutzt werden kann.

Abbildung 2.8 zeigt zwei Gebäudehypothesen, sowie die aus einer Stereozuordnung extrahierten 3D-Kanten. Offensichtlich wurde in dem Beispiel eine zu geringe Höhe für die Gebäude angenommen, die nun korrigiert / angepasst werden kann. Die Qualität der 3D-Gebäudehypothesen kann noch verbessert werden, indem zusätzliche Metainformationen einbezogen werden, die allerdings nicht in der ALK enthalten sind. Die Gebäudeparameter (Dachform, Dachhöhen, Firstrichtung) hängen auch von den Regeln der Stadtplaner ab. Solche zusätzlichen Bedingungen sind z.B.:

- Die Lage des Gebäudes zu einer Straße: heutzutage verläuft die Firstrichtung meist parallel zur Straßenrichtung, daher kann diese aus der Beziehung des Gebäudes zur Straße abgeleitet werden.
- Die Nachbarschaft zu anderen, nicht topologisch angrenzenden, Gebäuden: in den meisten Fällen besitzen benachbarte Gebäude (Häuserreihen) den gleichen Dachtyp, ähnliche Dachhöhe und gleiche Firstrichtung, da sie einem einheitlichen Bebauungsplan entstammen.
- Die Stadtlage des Gebäudes (Zentrum, Stadtrand, Industriegebiet) hat auch einen Einfluss auf die Dachform und Höhe eines Gebäudes: in Industriegebieten herrschen Flach- und Scheddächer vor. Am Stadtrand ist typischerweise eine 2-geschossige Einzelhausbebauung mit Sattel- oder Walmdach anzutreffen.

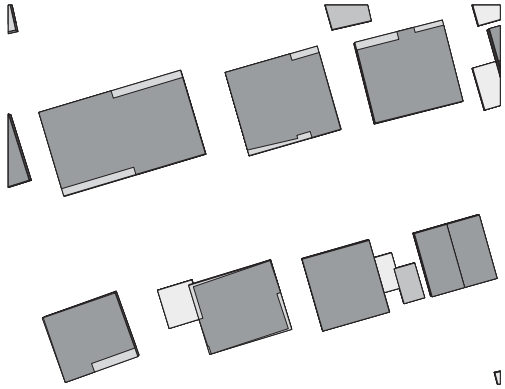


Abbildung 2.5: Generalisierung des Grundrisses durch minimal umschließendes Rechteck.

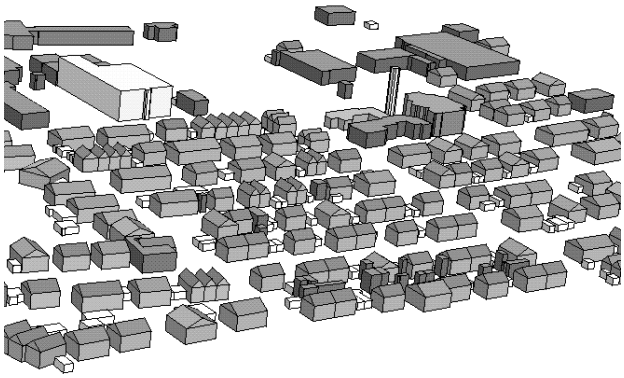


Abbildung 2.6: 3D-Gebäudehypothesen basierend auf den ALK-Gebäudegrundrissen.

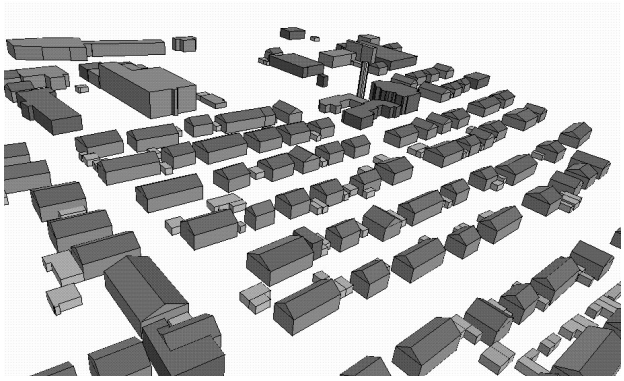


Abbildung 2.7: 3D-Gebäudehypothesen unter Berücksichtigung benachbarter Gebäude.

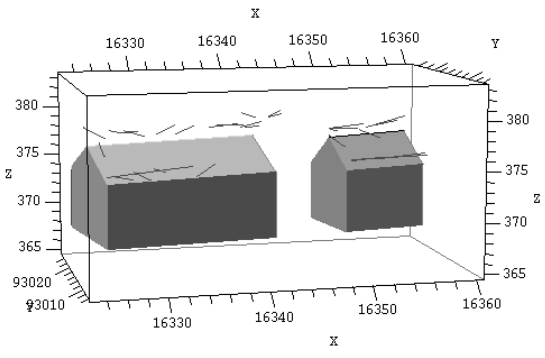


Abbildung 2.8: 3D-Ansicht der Gebäudehypothesen und zugeordneten DHM-Kanten

Solche Zusammenhänge können z.B. durch Statistiken aus bestehenden Datensätzen abgeleitet werden. Das Nachbarschaftsproblem könnte mit Hilfe von Clustering-Verfahren gelöst werden, die Cluster beliebiger Form erkennen können. Die so gefundenen Gebäudegruppen können dann weiter analysiert werden um z.B die Stadtlage des Gebäudes zu ermitteln. Im Kapitel 7 werden wir ein parameterfreies Clusterverfahren zur Gruppierung räumlicher Objekte beschreiben. Generell ermöglicht die Interpretation von bestehenden Datensätzen somit die automatische Ableitung von 3D-Stadtmodellen, die für viele Anwendungen eine hinreichend gute Approximation an die Realität darstellt (Abb. 2.9).

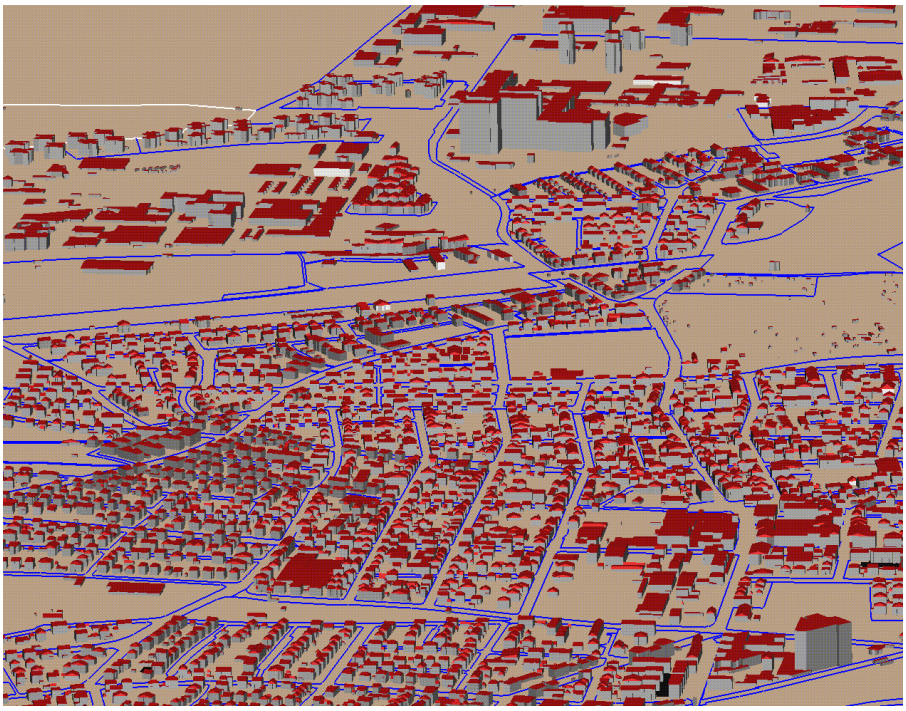


Abbildung 2.9: 3D-Gebäudehypothesen von Stuttgart Vaihingen Universität und Umgebung

2.2 Fortführung von ATKIS-Daten basierend auf ALK-Daten

Die Vermessungsverwaltungen des Bundes (BKG[1]) und der Länder (Landesvermessungsämter) gehören beide der AdV[2] an, innerhalb der sie gemeinsam am bundesweit einheitlichen Projekt $ATKIS$[3] arbeiten. Aufgabe des Projekts ATKIS ist es, die traditionellen topographischen Landeskartenwerke durch datenverarbeitungsfähige digitale Landschaftsmodelle zu ergänzen und diese digitalen Daten öffentlich-rechtlich bereitzustellen. Zu den topographischen Landeskartenwerken zählen die folgenden Karten:

- Topographische Grundkartenwerke in den Maßstäben 1 : 5.000 bis 1 : 10.000, dazu gehören die
  - Deutsche Grundkarte 1 : 5.000 (DGK5) und die
  - Topographische Karte 1 : 10.000 (TK10)
- Topographische Karte 1 : 25.000 (TK25)
- Topographische Karte 1 : 50.000 (TK50)
- Topographische Karte 1 : 100.000 (TK100)

[1] Bundesamt für Kartographie und Geodäsie (http://www.ifag.de).
[2] Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (http://www.adv-online.de).
[3] Amtliches Topographisch-Kartographisches Informationssytem (http://www.adv-online.de/produkte/atkis.htm)

- Topographische Übersichtskarte 1 : 200.000 (TÜK200)
- Übersichtskarte 1 : 500.000 (ÜK500)
- Bundesrepublik Deutschland 1 : 1.000.000 (D1000)
- Internationale Weltkarte 1 : 1.000.000 (IWK1000)

ATKIS ist als Geobasisinformationssystem gedacht, das öffentliche und private Nutzer mit geothematischen Fachdaten verknüpfen können, um raumbezogene Planungen und Visualisierungen durchzuführen. Ein weiteres wesentliches Ziel von ATKIS ist die Automatisierung der Erstellung der topographischen Landeskartenwerke, da die manuelle Erstellung und Fortführung topographischer Karten einen sehr zeit- und kostenintensiven Prozess darstellt. Die Grundidee dabei ist es, nicht mehr wie bisher jeden Maßstabsbereich für sich zu erfassen und fortzuführen, sondern nur noch wenige ausgewählte Maßstabsbereiche so zu modellieren und zu erfassen, damit andere Maßstäbe daraus automatisch abgeleitet werden können. Zu diesem Zwecke hat die AdV das Geobasisinformationssystem ATKIS in vier digitale Landschaftsmodelle (DLM) unterschiedlicher Maßstabsbereiche eingeteilt:

- **Basis-DLM**, Maßstab 1 : 25.000
- **DLM50**, Maßstab 1 : 50.000
- **DLM250**, Maßstabsbereich 1 : 200.000 bis 1 : 250.000
- **DLM1000**, Maßstab 1 : 1.000.000

Der Informationsgehalt des Basis-DLM orientiert sich an der topographischen Karte 1 : 25.000, da es als Grundlage zur Erzeugung der TK25 dienen soll. Die TK10 wird laut AdV jedoch auch aus dem Basis-DLM abgeleitet. Dies erscheint auf den ersten Blick widersprüchlich, jedoch muss man wissen, dass in vielen Bundesländern das Basis-DLM durch Digitalisierung der TK10 erfasst wurde und in manchen Bundesländern teilweise sogar durch Digitalisierung der DGK5. Der Grund für diese unterschiedlichen Erfassungen ist der Föderalismus der Bundesländer, wodurch ATKIS eine freiwillige Vereinbarung der Bundesländer ist, nach Möglichkeit Geobasisinformationen in einheitlicher Form zu erfassen. Ursprünglich wurde das Basis-DLM auch mit DLM25 bezeichnet, da aber ein Großteil der digitalen Datenbestände bereits genauer als für eine TK25 erfasst wurden, spricht man nun vom Basis-DLM. Vom Basis-DLM verlangt man nun, dass es alle nötigen Informationen für die Ableitung einer TK25 enthalten muss, wenn möglich jedoch den Informationsgehalt und die Lagegenauigkeit einer TK10. Das DLM50 soll als Datenbasis für die Erzeugung topographischer Karten im Maßstab von 1 : 50.000 bis 1 : 100.000 dienen und orientiert sich deshalb im Informationsgehalt an der TK50. Das DLM250 besitzt einen Informationsgehalt und eine Lagegenauigkeit entsprechend dem Maßstabsbereich 1 : 200.000 bis 1 : 250.000 und dient als Basis für die TÜK200 und die ÜK500. Das DLM1000 wiederum orientiert sich vom Informationsgehalt an der internationalen Weltkarte im Maßstab 1 : 1.000.000 und dient zur Erzeugung der D1000 und der IWK1000.

Wie oben bereits erwähnt ist das Basis-DLM häufig genauer erfasst als notwendig. Die Bestrebungen gehen jedoch in Zukunft dahin, dass nur noch ein einziges universelles Basis-DLM existiert, das die Erdoberfläche so genau modelliert, dass alle weiteren Landschaftsmodelle und daraus abgeleiteten Kartenwerke automatisch erzeugt werden können. Das würde bedeuten, dass nur noch ein einziges Modell manuell erfasst und fortgeführt werden muss. Da die ALK im Maßstab 1 : 1.000 erfasst wird (genauer gesagt ist 1 : 1.000 der Zielmaßstab, existierende Datensätze liegen im Bereich von 1 : 500 bis ca. 1 : 2.500) und Informationen, wie z.B. Flurstücksgrenzen, Flurstücksnutzung, Gebäudegrundrisse und Gebäudenutzung enthält, ergab sich die Frage, ob ALK-Daten dazu verwendet werden können, um geometrisch genaue Basis-DLM Daten abzuleiten. Diese Problemstellung gehört zum Aufgabengebiet der *kartographischen Modellgeneralisierung* und die Untersuchung grundlegender Möglichkeiten der automatischen Modellgeneralisierung ist im aktuellen Zusammenhang mit dem ATKIS DLM50 von besonderem Interesse, da es Ziel des BKG ist, das DLM50 vollständig automatisch aus dem Basis-DLM abzuleiten und nicht manuell erfassen zu müssen.

Die ALK beinhaltet Flurstücksgrenzen, Gebäudegrundrisse, Nutzungsarten und Straßennamen. Da der Prozess der Fortführung zeit- und kostenintensiv ist, entstand die Idee, ATKIS-Daten aus ALK-Daten abzuleiten, um den Aufwand an manueller Datenerfassung und Fortführung für das Basis-DLM zu verringern. Zur Zeit

werden beide Datenbestände getrennt voneinander erfasst und fortgeführt. Im folgenden wird aufgezeigt, dass es grundsätzlich möglich ist, ATKIS-Daten aus ALK-Daten abzuleiten. Dies wird anhand des Objektbereichs *Siedlung* verdeutlicht. Durch die Informationen über Grundstücksgrenzen, Nutzungsarten und den implizit gegebenen geometrischen und topologischen Informationen enthält die ALK alle notwendigen Informationen, die man für die Ableitung von Siedlungsobjekten benötigt. In Abbildung 2.10 ist der von uns verwendete ALK-Datensatz abgebildet.

Abbildung 2.10: Ausschnitt aus einem ALK-Datensatz

Die Ableitung von ATKIS-Objekten des Typs *Wohnbaufläche* (ATKIS-Objektart 2111[4]) aus ALK-Daten beinhaltet Generalisierungs- und Aggregationsoperationen, die im folgenden näher erläutert werden. Zur Beschreibung, wie räumliche Objekte in der ALK repräsentiert werden, benutzen wir die semantische Modellierung als eine konzeptionelle Methode zur Analyse, wie ein menschlicher Operateur diese Objekte in der Realität bzw. Luftbildern oder digitalen Datenbeständen wie der ALK erkennt.

2.2.1 Aggregations- und Generalisierungsoperatoren

Zur Ableitung neuer räumlicher Objekte benötigt man im allgemeinen Operatoren zur Aggregation und Generalisierung von räumlichen Objekten (Regnauld 1996, Molenaar 1996). Für die Aggregation räumlicher Objekte ist eine geeignete Definition der *Objektnachbarschaft* erforderlich, sowie ein Aggregationskriterium, z.B. ein gleiches Objektattribut. Wir haben dazu in unserem objektorientierten Datenmodell (Fritsch & Anders 1996) Operatoren für die Aggregation bezüglich *topologischer* und *metrischer* Nachbarschaft implementiert. Bei der Berechnung der topologischen Nachbarschaft werden die Adjazenz-Relationen im Datenmodell genutzt. Die metrische Nachbarschaft wird mithilfe von Abstandsmaßen berechnet. Wir verwendeten in diesem Fall dazu die sogenannte *Constrained Delaunay Triangulation* (Preparata & Shamos 1985) aller Objektpunkte und Objektlinien. Die Delaunay-Triangulation ermöglicht uns die Definition der räumlichen Nachbarschaft ohne explizite Angabe eines Abstandsmaßes, da diese Art der Triangulation Punkte nur mit ihren nächsten Nachbarn verbindet und existierende Kanten (vorhandene Objektlinien) mit berücksichtigt. In Kapitel 6 werden wir auf die verschiedenen Arten von *Nachbarschaftsgraphen* genauer eingehen. Für die Generalisierung der geometrischen

[4]Im neuen ALKIS-Objektartenkatalog, der zur Harmonisierung von ALK und ATKIS eingeführt wurde und in Zukunft die Objektarten von ALK und ATKIS einheitlich definieren wird, entspricht dies der Nummer 41001.

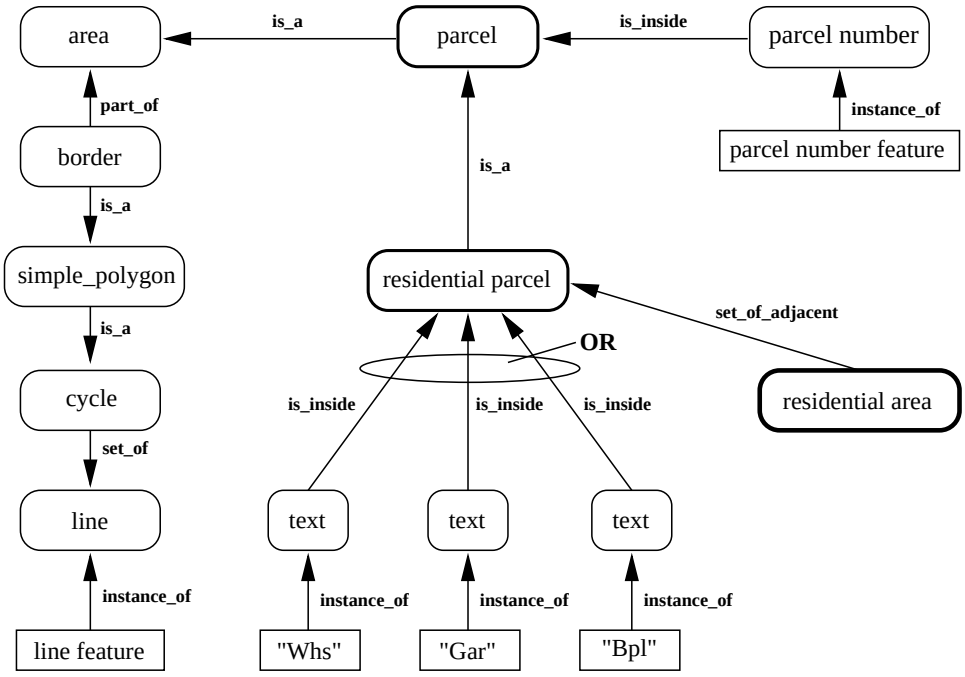


Abbildung 2.11: Semantisches Modell für Flurstücke und Wohnbauflächen

Form von gegebenen oder aggregierten räumlichen Objekten haben wir Operatoren zur Erzeugung der *konvexen Hülle* (Preparata & Shamos 1985) und des minimal umschließenden Rechtecks implementiert. Im folgenden Abschnitt beschreiben wir unseren Ansatz zur Ableitung von Wohnbauflächenobjekten aus ALK-Daten. Bei dieser Vorgehensweise wurde ausschließlich die topologische Nachbarschaft zur Aggregation räumlicher Objekte verwendet.

2.2.2 Ableitung der ATKIS-Objektart *Wohnbaufläche*

Gemäß dem ATKIS-Objektartenkatalog ergibt sich ein Siedlungsobjekt aus der Nutzung der in ihm enthaltenen Gebäudeobjekte. Der ALK-Datensatz bestand aus Linien-Elementen vom Typ Flurstücksgrenze, flächenhaften Gebäudeobjekten sowie punktförmigen Textobjekten, die eine Bezeichnung des Gebäudetyps bzw. der Flurstücksnummer oder Flurstücksnutzung (z.B. *Bpl*) enthielten. Der Zusammenhang zwischen ALK und ATKIS wird im folgenden in einem semantischen Modell hergestellt (vgl. Abb. 2.11).

1. Ein Siedlungsobjekt (*residential area*) ergibt sich durch Zusammenfassen angrenzender Flurstücke, die eine bestimmte Nutzung aufweisen (*residential, industrial, ...*).
2. Ein bezeichnetes Flurstückobjekt (*parcel*) ergibt sich aus einem Flurstück und der Angabe der Nutzung der auf ihm befindlichen Gebäudeobjekte.
3. Ein Flurstück ergibt sich durch Erzeugung geschlossener Polygone aus Linien-Elementen der Kategorie *Flurstücksgrenze*.

Der linke und obere Teil des Graphen in Abbildung 2.11 beschreibt den 3. Schritt, d.h., wie aus den Ausgangsdaten ein Flurstück gebildet wird. Im Datensatz lagen die Flurstücke nicht als Polygone, sondern lediglich als Sammlung von linienhaften Elementen und Textelementen vor (sogenannten Spaghetti-Daten)[5]. Deshalb ist der erste Schritt in unserem Ansatz, anhand der Linienobjekte und deren topologischen Beziehungen untereinander, alle möglichen geschlossenen Flächen zu berechnen. Eine Fläche wird in unserem Fall durch ein einfaches

[5]Die ALK selbst ist objektstrukturiert, jedoch lag uns nur ein unstrukturierter Testdatensatz im DXF-Format vor. Da aber unstrukturierte Daten häufiger vorkommen, beschreiben wir hier den allgemeinsten Fall der Interpretation von raumbezogenen Daten.

(a) Ermittelte Flurstücke (b) Abgeleitete Wohnbauflächen

Abbildung 2.12: Ergebnis der automtischen ALK-Daten-Auswertung

Polygon repräsentiert. Das einfache Polygon wiederum stellt sich als geschlossene Masche in dem durch die Linienobjekte aufgespannten Netz dar.

Im nächsten Arbeitsschritt bestimmen wir, in welchen von diesen Flächen eine Flurstücksnummer liegt. Dies wird durch einen Punkt-in-Polygon-Test für alle möglichen Kombinationen von Flächen und Flurstücksnummern erreicht. Durch eine raumbezogene Zugriffsstruktur kann der zeitliche Aufwand für solche geometrischen Operationen noch erheblich verringert werden. Das Ergebnis nach diesem Schritt ist in Abbildung 2.12 (a) dargestellt.

Nachdem alle Flurstücke ermittelt wurden, sind nun all diejenigen Flurstücke auszuwählen, die einem Wohnbauflächenbereich zugeordnet werden können. Dazu wählen wir aus allen vorhandenen Textsymbolen im ALK-Datensatz die Symbole *Whs, Gar, Bpl* (Wohnhaus, Garage, Bauplatz) aus, und führen wieder einen Punkt-in-Polygon-Test für alle möglichen Kombinationen dieser Symbole und den Flurstücksflächen aus.

Im letzten Arbeitsschritt wird eine Gruppierung aller extrahierten Wohnbauflächen vorgenommen, um ATKIS-Objekte des Typs Wohnbaufläche zu erstellen. Die semantische Regel für die Gruppierung solcher Flächen ist das Prinzip der Adjazenz (rechter Teil in Abb. 2.11). Mit anderen Worten, alle extrahierten Flächen, die eine gemeinsame Flurstücksgrenze besitzen, werden zusammengefasst. Diese Berechnung kann einfach ausgeführt werden, indem die topologischen Relationen benutzt werden, die aus unserem objektorientierten Datenmodell direkt abgeleitet werden können. Abbildung 2.12 (b) zeigt das Ergebnis dieser Gruppierung, überlagert mit den ALK-Daten.

Evaluierung des Ergebnisses

Das Ergebnis in Abbildung 2.12 (b) (hellgraue Flächen) wurde allein durch Aggregation angrenzender Wohnbauflächen nach dem semantischen Modell in Abbildung 2.11 ermittelt. Der Vergleich dieser automatisch erzeugten ATKIS-Objekte mit originalen ATKIS-Objekten (vgl. Abb. 2.14 (a) schwarze Linien) zeigt eine sehr gute Übereinstimmung, jedoch auch einige Unterschiede. Ursachen für diese Unterschiede sind:

- Die Grenzen der Wohnbauflächenbereiche stimmen deshalb nicht exakt überein, da in dem ATKIS-Datensatz die Straßen nicht als Fläche erfasst wurden, sondern durch ihre Mittelachsen repräsentiert werden. Da in ATKIS topologisch angrenzende Objekte durch die gleichen geometrischen Elemente dargestellt werden, treffen sich die Siedlungsgrenzen bei den Mittelachsen der Straßen. Unsere Grenzen entsprechen jedoch den genauen Grenzen zu den ALK-Straßenflächen. Man kann dieses Problem umgehen, indem man entweder gleichzeitig ATKIS-Straßenobjekte als Flächenobjekte aus den ALK-Daten ableitet oder einen Generalisierungsschritt einführt, der die Mittelachsen bestimmt und dann die Grenzen der Siedlungsflächen modifiziert. Man kann jedoch sagen, dass eine getrennte Ableitung von Siedlungsobjekten und Straßenobjekten nicht sinnvoll erscheint, sondern beide gemeinsam ermittelt werden müssen.

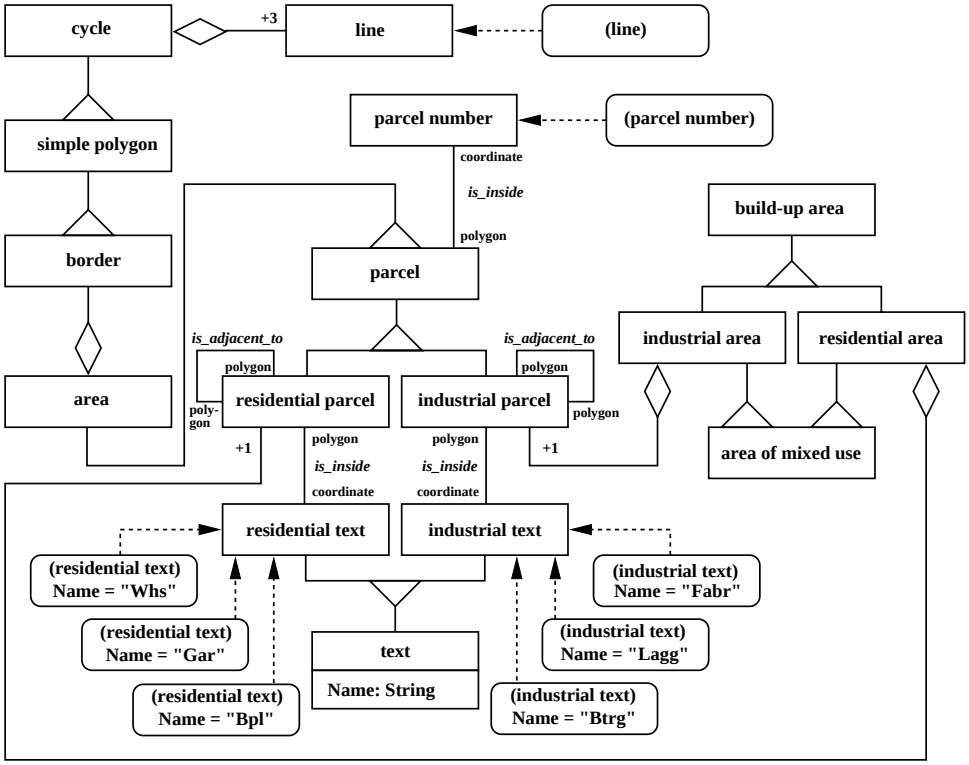


Abbildung 2.13: Erweitertes semantisches Modell für Flurstücke, Wohnbauflächen, Industrieflächen und Flächen mit gemischter Nutzung.

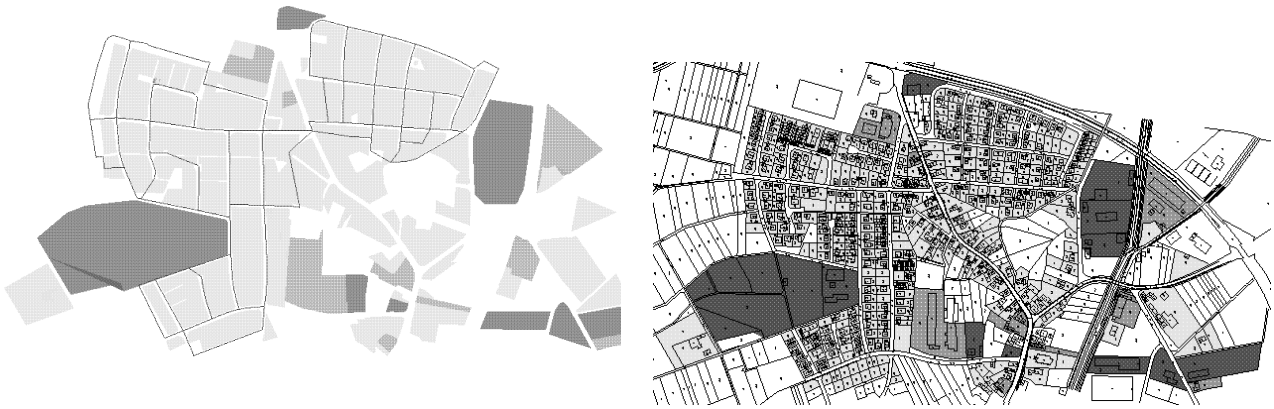
- Ein weiterer Unterschied entsteht dadurch, dass in unserem ersten Ansatz *Industrie- und Gewerbeflächen*, sowie *Flächen gemischter Nutzung* (Industrie- und Wohngebäude in einem Flurstück) nicht modelliert wurden. Die Flächen mit gemischter Nutzung wurden somit ebenfalls als Wohnbauflächen klassifiziert, was nicht vollständig falsch ist, jedoch in ATKIS einer eigenen Objektart zugeordnet wird. Wir erweiterten deshalb unser Modell, wie in Abbildung 2.13 dargestellt, um die Objekte Industrie- und Gewerbefläche (ATKIS-Objektart 2112 / ALKIS-Objektart 41002) und Fläche gemischter Nutzung (ATKIS-Objektart 2113 / ALKIS-Objektart 41007). Die Abbildung 2.14 (a) und (b) zeigen das verbesserte Ergebnis. Ein direkter Vergleich mit Abbildung 2.12 (a) zeigt, dass mit diesem Modell weniger Flächen als Wohnbaufläche klassifiziert werden.

Generell können auch

- verschiedene Erfassungszeiträume der ALK- und ATKIS-Daten,
- unterschiedliche Erfassungsregeln und
- die subjektive Interpretation des Operateurs bei der Erfassung der ATKIS-Daten aus Orthophotos

zu Unterschieden in den Ergebnissen führen.

Zusammenfassend kann man sagen, dass unsere Ergebnisse die Möglichkeit der Verwendung großmaßstäbiger Daten (z.B. ALK) zur automatischen Fortführung von Daten mittleren Maßstabs (z.B. ATKIS) bestätigen. Unser pragmatischer Ansatz zeigt, dass die Verknüpfung unterschiedlicher räumlicher Datenbestände effizient mit Hilfe eines semantischen Modells und räumlicher Aggregierungs- und Approximationsoperatoren durchführbar ist. Ein geeignetes semantisches Modell ist jedoch nicht immer vorhanden (keine Erfassungsregeln vorhanden) oder kann nur schwer erstellt werden (Verknüpfung der Erfassungsregeln und dem verwendeteten Datenbestand ist nicht direkt ersichtlich). Für solche Fälle können Methoden des *Maschinellen Lernens* (Michalski, Carbonell & Mitchell 1984, Sester 1995) genutzt werden. Durch die Methode *Lernen durch Beispiele* könnten prinzipiell automatisch semantische Konzepte oder Regeln abgeleitet werden. Eine andere Möglichkeit besteht darin,



(a) Überlagerung von original ATKIS-Wohnbauflächen (schwarze Linien)

(b) Überlagerung mit original ALK-Daten

Abbildung 2.14: Erweiterte Klassifizierung in: Wohnbaufläche (hellgrau), Gebiete gemischter Nutzung (grau) und Industrie- und Gewerbefläche (dunkelgrau) und Überlagerung mit original ATKIS- und ALK- Daten.

von einer sehr allgemeinen Regel auszugehen und beim Auftreten von Ausnahmen von dieser Regel, durch die Methode des *Inkrementellen Lernens*, das Verhalten des Systems zu verbessern.

Unser hier beschriebener Ansatz baut auf dem Flurstück als kleinste Flächeneinheit auf und gruppiert gleiche klassifizierte Flurstücke zusammen. Es ist jedoch auch möglich, die Gebäudegrundrisse der ALK nach der Nutzung zu gruppieren, um dann aus diesen Objektgruppen ATKIS-Siedlungsflächen abzuleiten, indem man z.B. die konvexe Hülle der gruppierten Gebäudegrundrisse als Siedlungsfläche interpretiert. Somit würde man Flächengeometrien erhalten, die nicht der Geometrie der Flurstücke folgt, sondern sich aus der Verteilung der Gebäude ergibt und möglicherweise einer besseren Interpretation der Daten für das ATKIS Basis-DLM entspricht. Jedoch erwartet ATKIS eine flächendeckende topologisch korrekte Modellierung der Flächen, die durch die Flurstücksgrenzen der ALK und den abgeleiteten Straßenmittelachsen aus Straßenflurstücken einfacher zu ermitteln ist als aus einer Menge topologisch unstrukturierter Flächen. *Data Mining* Methoden – speziell Clusterverfahren – können auch dann eingesetzt werden, wenn weder Vorinformationen noch geeignete Beispiele für zu detektierende Zusammenhänge in den Daten vorliegen.

Kapitel 3

Data Mining und Knowledge Discovery in Datenbanken

Moderne Datenbanktechnologien und Datenerfassungsmethoden haben dazu geführt, dass heutzutage sehr große Ansammlungen von Daten in digitaler Form existieren. Neben den klassischen Datenbankanwendungen (Banken, Versicherungen, Handel, etc.) werden besonders auf dem Gebiet der raumbezogenen Daten immer mehr Daten erfasst, da hier die digitale satelliten- und flugzeuggestützte Fernerkundung immer breitere Anwendung findet. Dieser enorme Anstieg an verfügbaren Daten macht es notwendig, Informationen (Wissen) aus vorhandenen Daten zu gewinnen. Es gibt zu viele Daten, aber zu wenig Informationen. Diese Problematik führte zur Entstehung des Themengebiets *Data Mining* oder *Knowledge Discovery in Databases*. In diesem Kapitel wird auf die wesentlichen Begriffe und Probleme des Data Mining im allgemeinen und des *Spatial Data Mining* im besonderen eingegangen, um einen Überblick über dieses Gebiet zu geben. Weiterführende Literatur zu diesem Thema findet man in (Piatetsky-Shapiro & Frawley 1991), (Holsheimer & Siebes 1994), (Fayyad, Piatetsky-Shapiro, Smyth & Uthurusamy 1996), (Weiss & Indurkhya 1998), (Anand, Bell & Hughes 1993),(Bell, Anand & Shapcott 1994),(Ester, Kriegel & Xu 1995),(Ester et al. 1996),(Fotheringham & Rogerson 1994), (Koperski & Han 1995),(Koperski, Adhikary & Han 1996),(Lu, Han & Ooi 1993),(Mohan & Nevatia 1988),(Molenaar 1996), (Ng & Han 1994),(Regnauld 1996),(Shaw & Wheeler 1994).

3.1 Data Mining

Bis heute gibt es keine adäquate deutsche Übersetzung für den englischen Begriff Data Mining. Wörtlich übersetzt steht Data Mining für Datenbergbau, wobei die Datenbank das Bergwerk darstellt und die neuen Informationen das zu fördernde Material. (Frawley, Piatetsky-Shapiro & Matheus 1991) geben folgende Definition: *Data Mining oder Knowledge Discovery in Databases ist ein Prozess zur Gewinnung interessanter (potentiell nützlicher), impliziter und vorher unbekannter Informationen aus großen Datenbanken.* (Weiss & Indurkhya 1998) definieren Data Mining als Suche nach wertvollen Informationen in sehr großen Mengen von Daten. Dabei arbeiten Mensch und Computer zusammen. Der Mensch entwirft die Datenbank, beschreibt Probleme und definiert Ziele. Der Computer sichtet die Daten und schaut nach Mustern, die die gesetzten Ziele erfüllen. Im allgemeinen kann man Data Mining als die Integration mehrerer Bereiche definieren. Dazu gehören im wesentlichen Maschinelles Lernen (Michalski et al. 1984), Datenbanken, Datenvisualisierung, Statistik und Informationstheorie.

An Data Mining Systeme werden häufig folgende Anforderungen gestellt:

- Effizienz
- Zuverlässigkeit
- Verarbeitung unterschiedlicher Datentypen
- Verarbeitung unterschiedlicher Datenquellen

Kategorie	Aufgabe
Vorhersage	Klassifikation Regression Zeitreihen
Wissensentdeckung	Trendanalyse Assoziationsregeln Generalisierung Clustering Visualisierung Datenbanksegmentierung Text Mining

Tabelle 3.1: Verschiedene Arten von Data Mining Problemen

- Interaktive Nutzung
- Komfortable Eingabe und Ausgabe
- Datenschutz und Datensicherheit

3.2 Data Mining Aufgaben und Methoden

(Weiss & Indurkhya 1998) teilen die typischen Data Mining Aufgaben in die zwei Kategorien *Vorhersage* und *Entdeckung von Wissen* ein (siehe Tabelle 3.1), wobei der Vorhersage das größere Gewicht beim Data Mining zufällt. Die Aufgabe der Vorhersage ist gegenüber der Entdeckung von Wissen genauestens definiert und ihre Ergebnisse sind potentiell brauchbarer. Der Begriff der Entdeckung von Wissen stellt dagegen einen Sammelbegriff für viele Themen dar, die in Beziehung zum Gebiet der Entscheidungsfindung, -unterstützung (Decision Support) stehen. Die Methoden, um die Aufgaben der Vorhersage und Wissensentdeckung zu lösen, überlappen sich häufig. Im folgenden werden die wesentlichsten Data Mining Aufgaben kurz beschrieben.

3.2.1 Vorhersage (Prediction)

Im wesentlichen gibt es zwei Arten der Vorhersage, die *Klassifikation* und die *Regression*. Vergangene Ereignisse werden anhand ihrer bekannten Auswirkungen untersucht und für zukünftige Fälle verallgemeinert.

Klassifikation

Bei der Klassifikation im allgemeinen handelt es sich darum, jeden Datensatz aus einer gegebenen Menge von Datensätzen anhand bestimmter Attributwerte (Klassifikationsmodell) einer Klasse von mehreren vordefinierten Klassen zuzuordnen. In Beziehung zum Data Mining kann die Klassifikation als *Lernen einer Abbildungsfunktion* definiert werden (Hand 1981), (Weiss & Kulikowski 1991), (McLachlan 1992). Ein einfaches Beispiel für die Anwendung von Klassifikationsmethoden ist die automatisierte Überprüfung der Kreditwürdigkeit von Bankkunden, anhand bekannter Parameter.

Regression

Die Regressionsanalyse kann wie die Klassifikation als Lernen einer Abbildungsfunktion definiert werden. Im Gegensatz zur Klassifikation wird ein Datum nicht einer bestimmten Klasse aus endlich vielen Klassen zugeordnet, sondern es wird eine reelle Zahl als Ergebnis geliefert. Anwendungen der Regression existieren viele, wie z.B. die Vorhersage der Menge an Biomasse in einem Waldgebiet aufgrund von Fernerkundungsdaten.

Analyse von Zeitreihen

Zeitreihenanalyse ist ein Sonderfall der Regression oder Klassifikation, wobei Messungen über die Zeit für die gleichen Attribute durchgeführt werden. Die Wettervorhersage oder Aktienkursanalyse sind Beispiele dafür.

3.2.2 Entdeckung von Wissen (Knowledge Discovery)

Aufgaben der Wissensentdeckung stellen gewöhnlicherweise eine Vorstufe zur Vorhersage dar, da im allgemeinen in dieser Vorstufe Informationen zur Vorhersage nur unzureichend vorhanden sind. Der Prozess der Wissensentdeckung stellt somit eine Ergänzung zur Vorhersage von Ereignissen aus Datenbeständen dar.

Trendanalyse / Abweichungserkennung

Das Ziel bei der Trendanalyse ist die Erkennung von signifikanten Veränderungen (Abweichungen) innerhalb einer Datenmenge von gegebenen Normwerten oder Werten, die bei vorangegangenen Messungen ermittelt wurden. Methoden zur Trendanalyse stehen in direktem Zusammenhang mit Methoden der klassischen Statistik. Die klassische Methode für die Änderungserkennung ist der sogenannte Signifikanztest.

Generalisierung / Zusammenfassung / Visualisierung

Bei der Datengeneralisierung wird eine große Menge von relevanten Daten von einer detaillierten Ebene auf eine abstrakte Ebene transformiert. Dazu gehören alle Methoden die versuchen, eine kompakte Beschreibung für eine Datenmenge zu finden. Der Mittelwert und die Standardabweichung einer Menge von reellen Zahlen stellt ein einfaches Beispiel für diesen Bereich dar. Da das Problem der Generalisierung / Zusammenfassung von Datenmengen in direkter Beziehung zu der Aufgabe der Visualisierung (graphischen Darstellung) von Daten steht, werden die Methoden zur zusammenfassenden Beschreibung von Datenmengen häufig bei der interaktiven Erkundung (interactive exploration) von Daten oder zur automatischen Reportgenerierung verwendet. Es existieren im wesentlichen zwei Ansätze: Der sogenannte *Data Cube* und die *attributorientierte Induktion*.

Assoziationsregeln

Eine spezielle Form der Generalisierung stellen die sogenannten *Assoziationsregeln* (Agrawal, Imielinski & Swami 1993, Agrawal & Srikant 1994) dar. Assoziationsregeln besitzen die folgende Form: $X_1 \wedge \cdots \wedge X_n \Longrightarrow Y_1 \wedge \cdots \wedge Y_m, \Theta$, wobei X_i und Y_i Attribute darstellen und Θ einen Konfidenzwert (z.B. 98% Sicherheit) repräsentiert.

Clustering

Die Aufgabe des Clusterings besteht darin, eine kompakte Beschreibung für eine Datenmenge zu finden, die auf einer endlichen Menge von Kategorien oder Clustern beruht (Titterington, Smith & Makov 1985), (Jain & Dubes 1988). Das Datenclustering kommt beim Data Mining immer dann zur Anwendung, wenn Datensätze ohne vordefinierte Klassifizierungsattribute gruppiert werden sollen. Die Gruppierung erfolgt bei dieser Methode nach der Regel: *Maximiere die Ähnlichkeit innerhalb einer Klasse und minimiere die Ähnlichkeit zwischen verschiedenen Klassen.* Im Zusammenhang mit dem Data Mining kann das Clustering den Methoden des unüberwachten Lernens zugeordnet werden. Beispiele für Clusterverfahren sind die unüberwachte Landklassifizierung aus Multispektraldaten oder das Auffinden von Kundengruppen in Marketing-Datenbanken.

Text Mining

In sogenannten *Data Warehouses* (große zentrale Datensammlungen in Firmen, die jegliche Art von Informationen enthalten) sind viele Arten von Informationen enthalten. Viele von diesen Informationen sind häufig in Form von Text vorhanden. Das kann von einzelnen Textfeldern bis zu kompletten Dokumenten reichen. Hierbei tritt das Problem auf, diese alphanumerischen Daten so aufzubereiten, dass sie mit den bekannten Data Mining Verfahren behandelt werden können (Weiss & Indurkhya 1998).

3.2.3 Data Mining Methoden

Im folgenden sind kurz die wesentlichen Methoden aufgelistet, die zur Lösung von Data Mining Aufgaben eingesetzt werden:

- Entscheidungsbäume
- Entscheidungsregeln
- Klassifikationsmethoden
- Nichtlineare Regression
- Statistik
- Beispielbasierte Methoden
- Induktive logische Programmierung
- Wahrscheinlichkeitsbasierte Abhängigkeitsgraphen

3.3 Raumbezogenes Data Mining

Im folgenden soll nur ein kurzer Überblick über das Gebiet des räumlichen Data Minings in Bezug auf geographische Informationssysteme gegeben werden. Koperski et al. (1996) geben einen ausführlichen Überblick zum Stand der Forschung und über zukünftige Entwicklungen auf dem Gebiet des *Spatial Data Mining*.

Spatial Data Mining ist die Erweiterung von Data Mining Methoden von Datenbanken ohne Raumbezug auf raumbezogene Datenbanken. Neben den Gebieten maschinelles Lernen, Statistik, Datenvisualisierung und Informationstheorie kommen nun die Gebiete der räumlichen Datenbanken (räumliche Indexstrukturen) und der algorithmischen Geometrie dazu. Techniken des Spatial Data Mining finden eine breite Anwendung in Geo-Informationssystemen (Bill & Fritsch 1991) und in der Fernerkundung. Diese Techniken können dazu verwendet werden, räumliche Daten zu verstehen, d. h. z. B. nach Beziehungen zwischen raumbezogenen Daten oder Beziehungen zwischen raumbezogenen und nicht raumbezogenen Daten zu suchen, raumbezogene Daten zu klassifizieren, Anfrageoptimierungen durchzuführen oder raumbezogene Wissensdatenbanken aufzubauen.

Die wesentlichen räumlichen Data Mining Aufgaben sind:

- **Räumliche Objekt Charakterisierung:** Erweiterung der Operation „Generalisierung / Zusammenfassung“ auf räumliche Datenbanken.
- **Räumliches Clustering** bezieht, gegenüber dem klassischen Clustering, die räumliche Nachbarschaft mit ein. Ein wesentliches Problem dabei ist, die räumliche Nachbarschaft in einer geeigneten Weise zu definieren.
- **Räumliche Assoziationsregeln** (Koperski & Han 1995) berücksichtigen zusätzlich räumliche Relationen. Dazu gehören topologische (Adjazenz, Überlappung, etc.) und geometrische (Abstand, Größe, etc.) Relationen.
- **Räumliche Muster:** Erweiterung der „Trendanalyse / Änderungserkennung“ auf räumliche Datenbanken. Dieses Gebiet ist besonders für räumlich temporale Datenbanken interessant. Ein räumliches Muster beschreibt die charakteristische Struktur von räumlich verteilten Objekten oder deren Veränderung. Eine Gruppe von Gebäuden entlang einer Straße kann z.B. eine lineare Struktur besitzen.

Im Bereich der Geo-Informationssysteme ergeben sich im wesentlichen folgende Aufgabenstellungen, die durch Spatial Data Mining Methoden unterstützt werden können:

Generalisierung
Das Problem der Generalisierung umfasst die sogenannte automatische Modellgeneralisierung zur Ableitung von topographischen Modellen für digitale Landschaftsmodelle unterschiedlicher Maßstäbe und die automatische kartographische Generalisierung zur Erstellung von gedruckten Karten.

Zusammenführen von Datenbeständen
Die Erfassung und Fortführung von Digitalen Landschaftsmodellen führt häufig zum Problem, dass zwei unterschiedliche Datenbestände einander zugeordnet werden müssen (Walter 1997). Dieses Map Matching oder *Conflation* genannte Problem erfordert eine geeignete Interpretation der räumlichen Datenbestände.

Spatial Reasoning
Landschaftsplaner und Stadtplaner benötigen geeignete Werkzeuge, um implizite räumliche Informationen wie z.B. Siedlungsstrukturen oder Stadtquartiere aus räumlichen Datenbanken ableiten zu können.

3.3.1 Räumliche Datenstrukturen, geometrische Algorithmen

In allen oben genannten Spatial Data Mining Aufgaben sind Operationen wie z.B. Berechnung der konvexen Hülle, Delaunay Triangulierung, Nächster Nachbar, räumliche Joins oder sogenannte Kartenverschneidungen notwendig. Deshalb werden effiziente räumliche Zugriffsmethoden und Datenstrukturen benötigt.

Räumliche Datenstrukturen bestehen aus Punkten, Linien, Flächen und Körpern und ihren topologischen Relationen untereinander. Um einen Zugriff auf räumliche Objekte über räumliche Koordinaten zu ermöglichen, werden mehrdimensionale Suchbäume verwendet. Beispiele für diese Indexstrukturen sind die *Quadranten Bäume* (Samet 1990), *k-d Bäume* oder *R-Bäume* (Güttman 1984). Im Fall des R-Baumes werden die Objekte durch umschreibende Rechtecke approximiert. Jeder Knoten des R-Baumes stellt ein umschreibendes Rechteck kleinerer Rechtecke dar und speichert somit eine bestimmte Anzahl anderer Rechtecke. Die Blätter enthalten einen Zeiger auf die Objektbeschreibung.

Räumliche Operationen, wie der räumliche Join oder geometrische Algorithmen, sind sehr aufwendige Operationen. In (Brinkhoff, Kriegel & Seeger 1993) wird ein Mehrebenen-Verfahren zur Berechnung des Spatial Joins vorgestellt, das auf dem R*-Baum basiert. Die Untersuchung von geometrischen Algorithmen gehört zum Gebiet des *Computational Geometry* (Preparata & Shamos 1985). Der Schwerpunkt in Bezug auf Spatial Data Mining liegt hier bei der Entwicklung effizienter Algorithmen, die mit externen Hintergrundspeichern arbeiten, um geometrische Probleme lösen zu können, die zu groß für den internen Speicher sind (Goodrich, Tsay, Vengroff & Vitter 1993).

3.3.2 Räumliche Nachbarschaft

Die räumliche Nachbarschaft zwischen Objekten ist für die Spatial Data Mining Aufgaben Aggregation, Approximation und Assoziation von grundlegender Bedeutung. Diese räumliche Relation läßt sich jedoch nicht eindeutig definieren, da ihre Definition von der Anwendung abhängt. Im allgemeinen kann man zwei Arten der räumlichen Nachbarschaft definieren:

Topologische Nachbarschaft
Die Nachbarschaft wird durch topologische Relationen zwischen den räumlichen Objekten definiert. In den meisten Fällen wird die *Adjazenz*-Relation verwendet, aber auch die Relationen *Schnitt* oder *Liegt_innerhalb* sind möglich (Egenhofer & Franzosa 1995).

Metrische Nachbarschaft (siehe auch Kapitel 6)
Die metrische Nachbarschaft bedarf der Definition eines Abstandsmaßes, wie z.B. der *Euklidischen Distanz* oder der *Manhattan Distanz*.

- **Distanz Puffer**
  Alle Objekte, die innerhalb eines gegebenen Abstands zu einem Objekt liegen, werden als dessen Nachbar betrachtet.
- **Nächster Nachbar**
  Nur das Objekt mit dem kleinsten Abstand wird Nachbar genannt. Die Angabe eines Abstandswertes ist nicht erforderlich.

- **Triangulation**
  Ein expliziter Abstandswert ist auch dann nicht erforderlich, wenn eine Triangulierung, wie z.B. die *Delaunay-Triangulation* (Preparata & Shamos 1985), zwischen den Objekten als Nachbarschaftsrelation definiert wird. In diesem Fall sind jedoch mehr als nur ein Nachbarobjekt möglich.

3.3.3 Spatial Data Mining Architekturen

Für das Problem des Data Mining wurden mehrere Architekturmodelle vorgeschlagen. Zu diesen Modellen gehören die Parallelarchitektur von (Holsheimer & Kersten 1994), der Data Mining Prototyp DBLEARN / DBMINER von (Han & Fu 1996) und die Multikomponenten-Architektur von (Matheus, Chan & Piatetsky-Shapiro 1993). Die meisten dieser Architekturen wurden auch für Spatial Data Mining verwendet oder erweitert. Insbesondere die Multikomponenten-Architektur wurde wegen ihrer Allgemeinheit von vielen anderen für Spatial Data Mining verwendet, wie z.B. (Ester et al. 1995). Dieses System besteht neben der **Datenbank**, einer **Wissensdatenbank** und einer sogenannten **Kontrolleinheit** aus folgenden vier Modulen: **Datenbankschnittstelle, Fokusierungsmodul, Musterextraktionsmodul** und **Evaluierungsmodul**. Eine Erweiterung für Spatial Data Mining betrifft die Datenbankschnittstelle, die eine raumbezogene Indexstruktur für den Zugriff auf die Daten bereitstellen muss. Die Wissensdatenbank ist um räumliche Konzepte zu erweitern und das Musterextraktionsmodul hat Data Mining Techniken in Verbindung mit geometrischen Algorithmen auszuführen, um raumbezogene Regeln und Relationen zu entdecken.

Kapitel 4

Clusteranalyse

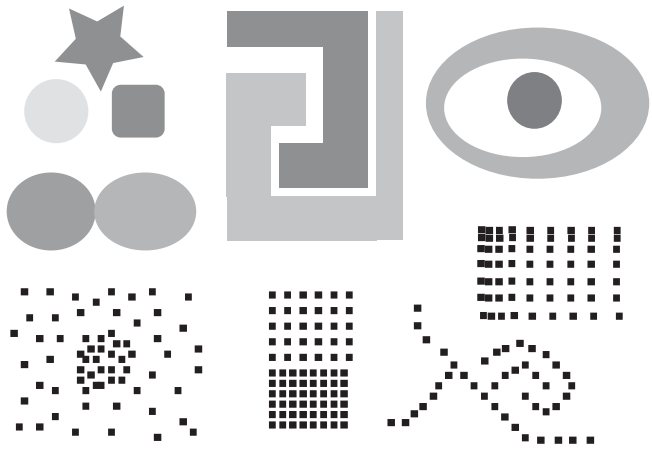


Abbildung 4.1: Clusterformen

Clustering ist eine bekannte Methode für die Dateninterpretation. Es bezeichnet den Prozess der Gruppierung von Objekten nach ihrer Ähnlichkeit. Typischerweise gibt man ein Ähnlichkeitsmaß vor und lernt mit Hilfe von statistischen Verfahren die optimale Zerlegung einer Datenmenge in Cluster. Clustering ist ein Hilfsmittel, um Informationen aus großen Datenbeständen – die u.U. durch viele Attribute pro Objekt gekennzeichnet sind – zu gewinnen. Die Daten können dabei beliebiger Natur sein, z.B. numerisch oder kategorisch, so lange man ein Ähnlichkeitsmaß definieren kann. Das Ziel des Clusterings ist es, auch versteckte Zusammenhänge im Datenbestand zu finden, daher sollten die Verfahren möglichst ohne spezifisches Vorwissen über den Datenbestand auskommen.

Das Ziel einer Clusteranalyse besteht darin, eine gegebene Menge von Daten in Cluster (Teilmengen, Gruppen, Klassen) einzuteilen, wobei die Einteilung folgende Eigenschaften besitzt:

- Homogenität innerhalb der Cluster, d.h. Daten, die demselben Cluster angehören, sollen möglichst ähnlich sein (Intra-Cluster-Homogenität).
- Heterogenität zwischen den Clustern, d.h. Daten, die unterschiedlichen Clustern zugeordnet sind, sollen möglichst verschieden sein (Inter-Cluster-Heterogenität).

D.h. Clusteralgorithmen gruppieren die Daten bzw. Objekte so in Klassen (Cluster), dass die Objekte einer Klasse ähnlich, die Objekte unterschiedlicher Klassen dagegen möglichst unähnlich sind. In Abbildung 4.1 gilt es z.B. die unterschiedlichen Objekte, die sich durch Farbe, Form und Dichte unterscheiden, zu identifizieren.

Typischerweise erfordert das Clustering Vorinformation, z.B. über die statistische Verteilung der Daten oder die Anzahl der zu detektierenden Cluster. Existierende Clusterverfahren wie k-means (Jain & Dubes 1988),

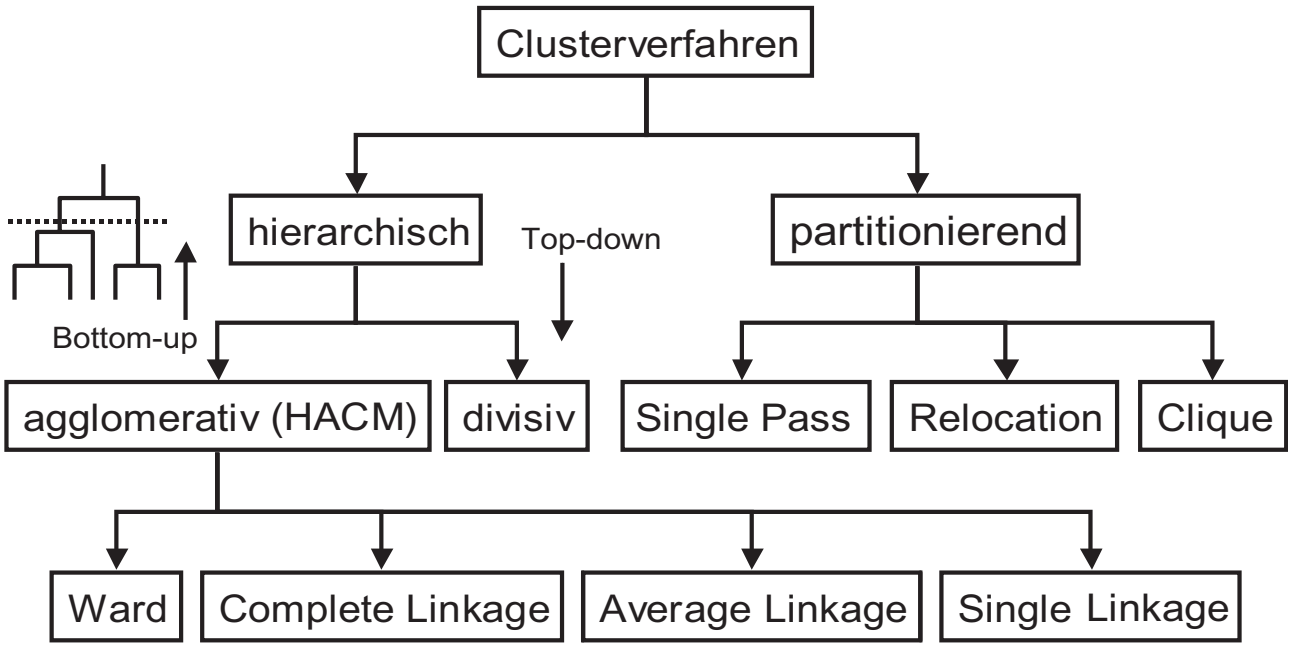


Abbildung 4.2: Clusterverfahren

PAM (Kaufman & Rousseeuw 1990), CLARANS (Ng & Han 1994), DBSCAN (Ester et al. 1996), CURE (Guha, Rastogi & Shim 1998) und ROCK (Guha, Rastogi & Shim 1999) basieren darauf, Cluster zu finden, die einem statistischen Modell genügen. K-means, PAM und CLARANS gehen von der Annahme aus, dass die Cluster hyperellipsoidisch, bzw. hypersphärisch sind und ähnliche Größen aufweisen. DBSCAN nutzt ein Dichte-basiertes Maß *density reachable*, welches alle Punkte eines Clusters erfüllen müssen, wohingegen Punkte, die zu unterschiedlichen Clustern gehören, diese Eigenschaft nicht aufweisen dürfen.

Alle diese Algorithmen hängen elementar von der Wahl der Parameter im statistischen Modell ab und können daher fehlschlagen, wenn diese nicht zur zugrundeliegenden Datenmenge passen, oder auch wenn das Modell die Charakteristika der Cluster (z.B. Form, Größe, Dichte) nicht korrekt beschreibt. Weiterhin müssen oft geeignete Abbruchkriterien für die Verfahren vorgegeben werden. Das heißt, diese Verfahren sind nicht parameterfrei. Wir werden im Kapitel 7 ein parameterfreies Verfahren beschreiben.

4.1 Methoden zur Clusteranalyse

Eine detaillierte Einführung in das Gebiet der Clusteranalyse findet man in (Berkhin 2002). Im folgenden wollen wir eine grobe Übersicht über existierende Clusterverfahren geben (Abb. 4.2). Clustering Algorithmen lassen sich wie folgt einteilen:

- Hierarchische Methoden
  - Agglomerative Verfahren
  - Divisive Verfahren
- Partitionierende Methoden
  - K-medoids Verfahren
  - K-means Verfahren
  - Dichte-basierte Verfahren
  - Relocation Verfahren

  - Probabilistische Verfahren
- Grid-basierte Methoden
- KI-Methoden
  - Neuronale Netze
  - Gradienten-Verfahren
  - Evolutionäre Verfahren
- Hochdimensionale Methoden
  - Projektive Verfahren
  - Clustering auf Unterräumen
- Skalierbare Algorithmen

Die Ergebnisse von Clusterverfahren können im wesentlichen wie folgt unterschieden werden:

- *Disjunktive* oder *nicht-disjunktive* Verfahren, d.h. die berechneten Cluster dürfen sich überlappen oder nicht.
- *Vollständige* oder *partielle* Verfahren, d.h. es werden alle gegebenen Objekte Clustern zugewiesen oder nur Teilmengen stellen Cluster dar.
- *Konvexe* oder *konkave* Verfahren, d.h. es sind nur Cluster mit konvexer Form möglich oder auch Cluster mit konkaver Form.

4.2 Hierarchisches und nicht-hierarchisches Clustering

4.2.1 Nicht-hierarchische Clusterverfahren

Nicht-hierarchische Clusterverfahren werden auch als partitionierende Cluster-Techniken bezeichnet. Es wird versucht, eine einfache Unterteilung der Daten in eine Menge von k nicht-überlappenden Clustern zu erreichen, wobei diese Unterteilungen ein vorgegebenes Kriterium optimieren. Jedes Cluster muss mindestens ein Element enthalten, und jedes Datenelement darf nur zu einer Gruppe gehören. Die meisten partitionierenden Verfahren gehen von einer vorgegebenen Startunterteilung aus; anschließend wird die Zugehörigkeit der Datenelemente im Laufe der Iterationen sukzessive adaptiert, um eine bessere Unterteilung zu erreichen. *Zentroid-basierte Verfahren* wie die k-means-Methode (MacQueen 1967), (Jain & Dubes 1988) und der ISODATA-Algorithmus (Ball & Hall 1965) weisen die Datenelemente denjenigen Clustern zu, deren mittlere Euklidische Distanz zum Clusterzentrum minimal ist. Diese Verfahren sind nur für metrische Räume geeignet, da sie den Zentroid (Mittelpunkt) aus einer gegebenen Menge von Datenelementen ermitteln müssen. *Medoid-basierte Verfahren* wie CLARANS und PAM nutzen ein repräsentatives Datenelement, den sogenannten Medoid, und suchen die Summe der Abstände zwischen dem Medoid und den ihm zugeordneten Datenelementen zu minimieren.

Ein Nachteil der Zentroid- und Medoid-basierten Verfahren ist, dass nicht alle Vorgaben von k zu natürlichen Clustern führen. Daher müssen die Verfahren in der Regel mehrfach durchlaufen werden, um die beste Unterteilung zu erhalten. Diese Entscheidung kann mittels vorgegebener Optimierungskriterien automatisiert werden. Der größte Nachteil liegt allerdings darin, dass sie lediglich konvexe Clusterformen ermitteln können: Konkave Formen, bei denen es vorkommt, dass ein Datenelement näher dem Repräsentanten eines anderen Clusters liegt, als dem eigenen, können nicht erkannt werden. Diese Formen kommen jedoch in natürlichen Clustern häufig vor; auch können natürliche Cluster oft von stark unterschiedlicher Größe sein.

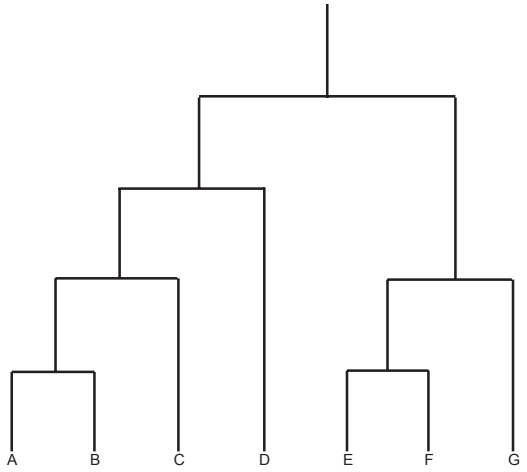


Abbildung 4.3: Beispiel für ein Dendrogramm.

4.2.2 Hierarchische Clusterverfahren

Hierarchische Clusterverfahren erzeugen ein Dendrogramm (Abb. 4.3), d.h. eine Baumstruktur, die eine Abfolge von verknüpften Clustern darstellt. Diese Abfolge beschreibt unterschiedliche Ebenen der Unterteilung. An der Wurzel des Baumes befindet sich ein einziges Cluster, welches alle anderen beinhaltet. In den Blättern des Baumes befinden sich Cluster, die aus nur einem Datenelement bestehen. Dendrogramme können sowohl top-down als auch bottom-up erzeugt werden. Die bottom-up Methode, auch als *agglomerative Technik* bezeichnet, startet mit jedem Datenelement als Cluster. In jedem Schritt des agglomerativen Algorithmus werden die zwei ähnlichsten Cluster zusammengefasst – wobei natürlich ein Ähnlichkeitsmaß vorgegeben sein muss. Die Anzahl der Cluster reduziert sich mit jedem Aggregationsschritt um eins. Das Verfahren wird so lange iteriert, bis entweder ein einziges großes Cluster übrig geblieben ist, eine vorgegebene Anzahl von Clustern erzeugt wurde, oder der Abstand zweier Cluster über einem vorgegebenen Schwellwert liegt. Die top-down-Methode arbeitet umgekehrt – sie wird auch als *divisives Verfahren* bezeichnet. Die agglomerativen Verfahren sind in der Literatur am meisten verbreitet.

Es existieren sehr viele unterschiedliche Varianten von hierarchischen Algorithmen. Generell arbeiten hierarchische Clusterverfahren auf Basis einer Ähnlichkeitsmatrix, wobei jedes Element der Matrix die Ähnlichkeit zwischen zwei Datenelementen beschreibt. In jedem Schritt des Algorithmus muss die Ähnlichkeitsmatrix aufdatiert werden, um die Änderungen, die sich durch die Cluster-Zusammenfassung ergeben, fortzuführen. Die Verfahren lassen sich folgenden drei Techniken zuordnen:

- *Zentroid (bzw. Medoid)* basierte Verfahren,
- *Linkage* basierte Techniken und
- *Varianz (bzw. Fehlerquadratsummen)* basierte Verfahren.

Das erste Verfahren hat ähnliche Eigenschaften wie die nicht-hierarchischen Verfahren (z.B. k-means und k-medoid), d.h. es kann keine beliebigen Clusterformen detektieren.

Das älteste Linkage-Verfahren ist der *single linkage Algorithmus*, der manchmal auch als Nächste-Nachbar-Verfahren bezeichnet wird. Dieses Verfahren benötigt keinen Repräsentanten, sondern das Cluster wird durch alle seine Datenelemente repräsentiert. Der Abstand zwischen zwei Clustern ergibt sich dabei aus dem Abstand jeweils zweier Datenelemente aus den unterschiedlichen Clustern. Im Falle des single linkage Verfahrens wird als Abstandsmaß die kürzeste Distanz verwendet. Weitere Verfahren ergeben sich durch Wahl eines anderen Distanzmaßes:

- single linkage Verfahren: $d(I, J) := \min_{x \in I, y \in J} d(x, y)$

- complete linkage Verfahren: $d(I,J) := \max_{x \in I, y \in J} d(x,y)$
- average linkage Verfahren: $d(I,J) := \frac{1}{|I||J|} \sum_{x \in I, y \in J} d(x,y)$
- Ward Verfahren: $d(I,J) := \frac{2|I||J|}{|I|+|J|} d(\mu_I, \mu_J)$

Die Linkage-Methoden können Cluster beliebiger Form und Größe ermitteln. Der Nachteil liegt jedoch darin, dass sie nicht robust gegenüber Rauschen und Ausreißern in den Daten sind. Weiterhin ergeben sich Probleme bei nur schwach separierbaren Clustern.

Um die Nachteile der hierarchischen Verfahren zu überwinden, wurden Algorithmen wie die *Methode der Shared-Near-Neighbors* (Jarvis & Patrick 1973), bzw. CURE (Guha et al. 1998) und ROCK (Guha et al. 1999) vorgeschlagen. Anstelle eines einzigen Cluster-Repräsentanten nutzt CURE eine vorgegebene konstante Anzahl an Repräsentanten für ein Cluster. Der ROCK Algorithmus arbeitet auf einem abgeleiteten Ähnlichkeitsgraphen und berücksichtigt ein vorgegebenes Modell für die Inter-Cluster-Heterogenität. Die Methode der Shared-Near-Neighbors nutzt einen *k-Nächste-Nachbar-Graph*, um die Ähnlichkeit zwischen zwei Clustern zu bestimmen. Der Vorteil dieses Verfahrens gegenüber den meisten anderen ist, dass es ohne einen vorgegebenen absoluten Maßstab auskommt, um die Ähnlichkeit (Nachbarschaft) von Objekten zu bestimmen. Karypis, Han & Kumar (1999) beschreiben ein Verfahren, das ebenfalls auf dem k-Nächste-Nachbar-Graphen aufbaut.

4.3 Graphbasiertes Clustering

Nach (Jaromczyk & Toussaint 1992) sind graphbasierte Clusterverfahren die mächtigsten Werkzeuge, um Ergebnisse zu erzielen, die dem menschlichen Leistungsvermögen nahe kommen. Die grundlegende Idee der graphbasierten Clusterverfahren ist sehr einfach: Aus den Originaldaten wird ein Nachbarschaftsgraph berechnet (z.B. der Minimal Spannende Baum). In diesem Graphen werden diejenigen Kanten eliminiert, die – gemäß einem vorzugebenden Kriterium – länger sind als ihre Nachbarn. Das Ergebnis ist schließlich ein Graphengeflecht (Wald), in dem jeder Baum ein Cluster repräsentiert. Die sogenannten *Baumzerlegungsverfahren* (van Schröder 2001) gehören ebenfalls zu den graphbasierten Verfahren. Sie setzen jedoch voraus, dass es sich bei dem Graphen um einen Baum (azyklischen Graphen) handelt. Das in dieser Arbeit entwickelte Verfahren (Kap. 7) gehört auch zu den graphbasierten Clusterverfahren. Im folgenden Kapitel werden wir auf die Nachbarschaftsgraphen genauer eingehen.

Kapitel 5

Ähnlichkeits- und Distanzmaße

Wie wir im vorangegangenen Kapitel gesehen haben, benötigt jedes Clusterverfahren ein Maß, um die *Ähnlichkeit* zwischen einzelnen Elementen des betrachteten Objektraums, zwischen Gruppen von Elementen (Clustern) oder zwischen Elementen und Clustern bestimmen zu können. Ein solches Maß ist nicht trivial, da im allgemeinen in Datenbanken mehr nicht-numerische Daten als numerische gespeichert sind. Lassen sich die nicht-numerischen Daten eindeutig auf Zahlen abbilden, kann das Clustering-Problem auf einen $n-$dimensionalen Vektorraum abgebildet werden, in dem dann eine klassische Metrik aus der Mathematik zur Messung der Distanz oder Ähnlichkeit verwendet werden kann. Der Vergleich zwischen Clustern lässt sich aber meistens nur durch zusätzliche Bedingungen geeignet definieren. Zuerst werden wir in diesem Kapitel auf die möglichen verschiedenen Arten (Skalen) von Daten eingehen und dann die Definition von Ähnlichkeits- und Distanzmaßen erläutern. Bei der folgenden Aufzählung von Maßen erheben wir natürlich keinen Anspruch auf Vollständigkeit. Es soll einzig gezeigt werden, wie vielfältig das Problem des Vergleichs von Objekten ist und immer im Kontext zu einem Modell definiert werden muss.

5.1 Skalentypen

Im allgemeinen sind die Merkmale eines zu gruppierenden Datensatzes von unterschiedlicher Art, d.h. die Merkmale besitzen nicht den gleichen *Skalentyp* (Abb. 5.1). Skalentypen lassen sich in *quantitative* und *qualitative* Typen einteilen.

Die quantitativen Typen entsprechen den Zahlen oder Typen, die sich eindeutig auf eine Zahlenmenge abbilden lassen. Mit quantitativen Daten kann man somit *rechnen* (+,-,*,/), sie lassen sich *anordnen/vergleichen* und „vermessen“, da die Zahlenmengen *metrische* Räume sind. Die quantitativen Daten lassen sich in *diskrete* und in *kontinuierliche* Typen unterteilen. Die diskreten Typen lassen sich auf die Menge der natürlichen Zahlen abbilden. Die kontinuierlichen Typen lassen sich dagegen nur auf Teilmengen der reellen Zahlen abbilden. Die Anzahl von Besuchern einer Veranstaltung sind ein Beispiel für einen diskreten Skalentyp. Die Fläche von Grundstücken, gemessen in m^2, ist dagegen ein Beispiel für einen kontinuierlichen Skalentyp.

Qualitative Daten sind alle Arten von Daten mit denen man nicht rechnen kann. Im günstigsten Fall lassen sich qualitative Daten in eindeutiger Weise anordnen, wie die sogenannten *ordinalen* Datentypen (z.B. die Körpergröße mit der Wertemenge {*klein, mittel, groß*}). Die qualitativen Daten lassen sich in die schon erwähnten ordinalen Typen und in die sogenannten *nominalen* und *binären* Typen einteilen. Mit nominal bezeichnet man *mehrkategoriale* Daten, wie z.B. die Haarfarbe mit der Wertemenge $\{schwarz, rot, braun, blond\}$, d.h. man kann diesen Merkmalen verbale Ausprägungen zuordnen und diese auf Gleichheit überprüfen, jedoch kann man weder mit ihnen rechnen noch lassen sie sich in eindeutiger Weise anordnen. Binäre Skalentypen[1] kann man als Sonderfall der nominalen Skalentypen auffassen, da sich ebenfalls für die Merkmale nur die Gleichheit überprüfen lässt. Zusätzlich besteht bei binären Skalentypen die Wertemenge aus nur zwei Elementen, wie z.B. $\{ja, nein\}$, $\{gut, schlecht\}$ oder {*männlich,weiblich*}.

[1] Binäre Skalentypen werden auch als *dichotome* Variablen bezeichnet.

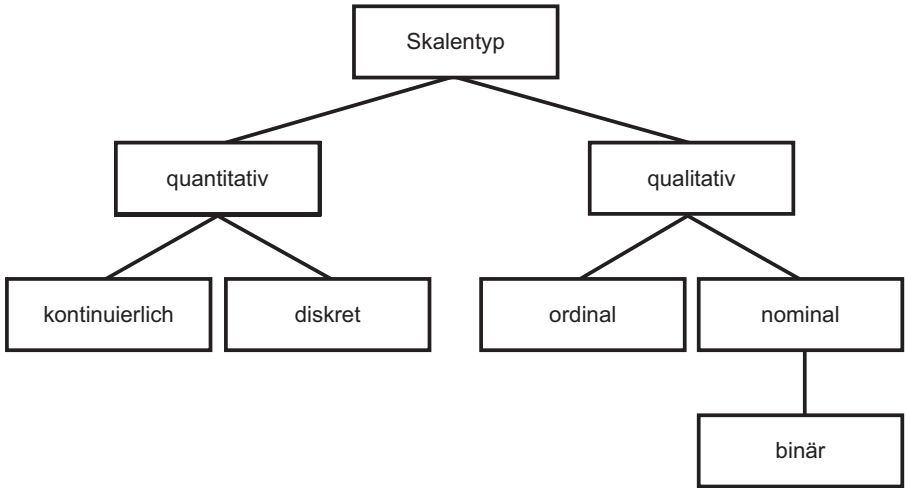


Abbildung 5.1: Skalentypen

5.2 Ähnlichkeit

Ähnlichkeit sollte sich nach Möglichkeit als kontinuierlicher Wert zwischen *identisch* und *vollständig unterschiedlich* messen lassen und kann formal wie folgt definiert werden:

Definition 5.2.1 (Ähnlichkeitsmaß)
Sei $M = \{m_1, \ldots, m_n\}$ *eine Menge von Merkmalen, dann nennt man eine Funktion* $s : M \times M \rightarrow [0, 1]$ Ähnlichkeitsfunktion, Ähnlichkeitsmaß oder auch Ähnlichkeitskoeffizient*, wenn für* $x, y \in M$ *gilt:*

$$\begin{array}{lcll} (S1)\ s(x,y) & = & s(y,x) & : \textit{Symmetrie} \\ (S2)\ s(x,y) & = & 1 \Leftrightarrow x = y & : \textit{Identität} \end{array}$$

D.h. Ähnlichkeit ist zum einen symmetrisch (S1), zum anderen zeigt sich eine maximale Übereinstimmung der Elemente dann, wenn $s = 1$ wird (S2). Von dieser Definition ausgehend gibt es eine Vielzahl von Ähnlichkeitsmaßen, von denen im folgendem eine Auswahl beschrieben wird.

5.2.1 Ähnlichkeitsmaße für binäre Skalentypen

Im Falle rein binärer Merkmalsvektoren, d.h. Vektoren deren Merkmale jeweils nur zwei Zustände einnehmen, die wir der Einfachheit wegen mit 0 und 1 bezeichnen wollen, kann man beim Vergleich die Anzahl der Merkmale mit gleichem und ungleichem Wert bestimmen. Man unterscheidet für zwei Merkmalsvektoren unter den folgenden vier Summen:

- $\sum_{\mathbf{00}}$: Anzahl gleicher Merkmale von x und y mit dem Wert 0.
- $\sum_{\mathbf{01}}$: Anzahl ungleicher Merkmale von x und y mit dem Wert 0 für x und dem Wert 1 für y.
- $\sum_{\mathbf{10}}$: Anzahl ungleicher Merkmale von x und y mit dem Wert 1 für x und dem Wert 0 für y.
- $\sum_{\mathbf{11}}$: Anzahl gleicher Merkmale von x und y mit dem Wert 1.

Aufbauend auf diesen vier Summen lassen sich die folgenden Ähnlichkeitsmaße definieren.

M-Koeffizient

Definition 5.2.2 (M-Koeffizient)

$$s(x,y) = \frac{\sum_{00} + \sum_{11}}{\sum_{00} + \sum_{01} + \sum_{10} + \sum_{11}}$$

S-Koeffizient

Definition 5.2.3 (S-Koeffizient)

$$s(x,y) = \frac{\sum_{11}}{\sum_{11} + \sum_{01} + \sum_{10}}$$

Tanimoto/Rogers-Koeffizient

Definition 5.2.4 (Tanimoto/Rogers-Koeffizient)

$$s(x,y) = \frac{\sum_{00} + \sum_{11}}{\sum_{00} + \sum_{11} + 2(\sum_{01} + \sum_{10})}$$

5.2.2 Ähnlichkeitsmaße für nominale Skalentypen

Verallgemeinerter M-Koeffizient

Für nicht binäre nominale Merkmalsvektoren lässt sich der M-Koeffizient verallgemeinern, indem man einfach das Verhältnis der Anzahl gleicher Merkmale zu der Anzahl der Merkmale als Ähnlichkeitsmaß definiert.

Definition 5.2.5 (Verallgemeinerter M-Koeffizient)

$$s(x,y) = \frac{\sum(GleicheMerkmale)}{\sum(Merkmale)}$$

5.2.3 Beispiel für binäre und nominale Skalentypen

Als Beispiel verwenden wir drei Gebäude mit unterschiedlichen Eigenschaften, die in Tabelle 5.1 aufgelistet sind. Die Tabelle 5.2 zeigt die zugehörigen M-, S- und Tanimoto/Rogers-Koeffizienten, nach denen Haus 1 und 3 sich am ähnlichsten sind.

Merkmale	Haus 1	Haus 2	Haus 3
Parkett	nein	ja	nein
Terrasse	ja	ja	nein
Balkon	nein	ja	ja
Niedrigenergiebauweise	ja	nein	ja

Tabelle 5.1: Beispiel für binäre Merkmale.

Im Falle einer nicht-binären Nominalskala sind diese Koeffizienten jedoch mit Vorsicht zu verwenden. Als Beispiel soll hier die Dachform eines Gebäudes dienen. Eine Möglichkeit zum Vergleich von nicht-binären Variablen ist

Ähnlichkeitsmaß	Haus 1-2	Haus 1-3	Haus 2-3
M-Koeffizient	1/4	1/2	1/4
S-Koeffizient	1/4	1/3	1/4
Tanimoto-Koeffizient	1/7	1/3	1/7

Tabelle 5.2: Ähnlichkeitsmaße zu Tabelle 5.1.

es, diese Variable als binären Vektor aller möglichen Ausprägungen zu betrachten (Tab. 5.3) und dann, wie im binären Fall, den M- S- oder Tanimoto/Rogers-Koeffizient zu berechnen. Nach Tabelle 5.3 ist es offensichtlich, dass Haus 1 und 2 (bzgl. Dachform) eine Ähnlichkeit von 1 haben und in den beiden anderen Fällen eine Ähnlichkeit von 0 besteht. Wie man jedoch in Tabelle 5.4 sieht, liefern der M- und Tanimoto/Rogers-Koeffizient in diesen Fällen eine Ähnlichkeit ungleich 0. Nur der S-Koeffizient und der verallgemeinerte M-Koeffizient liefern hier das erwartete Ergebnis. Der verallgemeinerte M-Koeffizient macht jedoch erst im Falle mehrerer nicht-binärer Variablen Sinn (siehe Tabelle 5.5 und 5.6).

Dachform	Haus 1	Haus 2	Haus 3
Satteldach	nein	nein	ja
Flachdach	ja	ja	nein
Walmdach	nein	nein	nein
Pultdach	nein	nein	nein

Tabelle 5.3: Beispiel für ein nominales Merkmal.

Ähnlichkeitsmaß	Haus 1-2	Haus 1-3	Haus 2-3
M-Koeffizient	1	1/2	1/2
S-Koeffizient	1	0	0
Tanimoto-Koeffizient	1	1/3	1/3
Verallg. M-Koeffizient	1	0	0

Tabelle 5.4: Ähnlichkeitsmaße zu Tabelle 5.3.

Die Interpretation einer nicht-binären nominalen Variablen macht jedoch Sinn, wenn man binäre und nicht-binäre Merkmale zusammen betrachtet. Die Tabellen 5.7 und 5.8 zeigen ein Beispiel für diesen Fall. Das Problem bei dieser Vorgehensweise ist jedoch, dass im Vergleich zu dem „binarisierten“ nominalen Merkmal (Dachform), die Ähnlichkeit aufgrund des *Nicht-Vorhandenseins* der binären Merkmale (Parkett, Terrasse, etc.) verloren geht. Diese Art der Verzerrung kann man ausgleichen, indem man, wie in Tabelle 5.9 dargestellt, auch für die binären Variablen den inversen (negierten) Fall berücksichtigt. Die Ähnlichkeitsmaße ändern sich dann, wie in Tabelle 5.10 dargestellt.

5.2.4 Ähnlichkeitsmaße für quantitative Skalentypen

Korrelationskoeffizient

Bei der Analyse von quantitativen Daten kommt es häufig vor, dass man bei einer Stichprobe gleichzeitig zwei Eigenschaften (Merkmale) erfasst, wie z.B. die monatliche Sonnenscheindauer vormittags und nachmittags oder der Kopfumfang und die Körperlänge von Neugeborenen. Möchte man nun feststellen, ob ein direkter Zusammenhang zwischen beiden Größen besteht, bietet es sich an, den sogenannten Korrelationskoeffizienten beider Größen zu bestimmen.

Merkmale	Haus 1	Haus 2	Haus 3	Haus 4
Farbe	weiß	rot	weiß	blau
Heizung	Gas	Öl	Solar	Gas
Dachform	Flachdach	Satteldach	Pultdach	Flachdach

Tabelle 5.5: Beispiel für gemeinsame Verwendung von nominalen und binären Merkmalen.

Merkmale	Verallgemeinerter M-Koeffizient
Haus 1 - 2	$0/3 = 0$
Haus 1 - 3	$1/3 \approx 0,33$
Haus 1 - 4	$2/3 \approx 0,66$
Haus 2 - 3	$0/3 = 0$
Haus 2 - 4	$0/3 = 0$
Haus 3 - 4	$0/3 = 0$

Tabelle 5.6: Verallgemeinerter M-Koeffizient zu Tabelle 5.5.

Merkmale	Haus 1	Haus 2	Haus 3
Parkett	nein	ja	nein
Terrasse	ja	ja	nein
Balkon	nein	ja	ja
Niedrigenergiebauweise	ja	nein	ja
Dachform: Satteldach	nein	nein	ja
Dachform: Flachdach	ja	ja	nein
Dachform: Walmdach	nein	nein	nein
Dachform: Pultdach	nein	nein	nein

Tabelle 5.7: Beispiel 1 für gemeinsame Verwendung von nominalen und binären Merkmalen.

Ähnlichkeitsmaß	Haus 1-2	Haus 1-3	Haus 2-3
M-Koeffizient	$5/8 = 0,625$	$1/2 = 0,5$	$3/8 = 0,375$
S-Koeffizient	$2/5 = 0,4$	$1/5 = 0,2$	$1/6 \approx 0,167$
Tanimoto-Koeffizient	$5/11 \approx 0,455$	$1/3 \approx 0,333$	$3/13 \approx 0,231$

Tabelle 5.8: Ähnlichkeitsmaße zu Tabelle 5.7.

Merkmale	Haus 1	Haus 2	Haus 3
Parkett: ja	nein	ja	nein
Parkett: nein	ja	nein	ja
Terrasse: ja	ja	ja	nein
Terrasse: nein	nein	nein	ja
Balkon: ja	nein	ja	ja
Balkon: nein	ja	nein	nein
Niedrigenergiebauweise: ja	ja	nein	ja
Niedrigenergiebauweise: nein	nein	ja	nein
Dachform: Satteldach	nein	nein	ja
Dachform: Flachdach	ja	ja	nein
Dachform: Walmdach	nein	nein	nein
Dachform: Pultdach	nein	nein	nein

Tabelle 5.9: Beispiel 2 für gemeinsame Verwendung von nominalen und binären Merkmalen.

Ähnlichkeitsmaß	Haus 1-2	Haus 1-3	Haus 2-3
M-Koeffizient	$1/2 = 0,5$	$5/12 \approx 0,42$	$1/3 \approx 0,33$
S-Koeffizient	$1/4 = 0,25$	$1/4 = 0,25$	$1/9 \approx 0,11$
Tanimoto-Koeffizient	$1/3 \approx 0,33$	$5/29 \approx 0,17$	$1/5 = 0,2$

Tabelle 5.10: Ähnlichkeitsmaße zu Tabelle 5.9.

Definition 5.2.6 (Korrelationskoeffizient)
Sei eine Stichprobe aus einer zweidimensionalen XY-Grundgesamtheit mit den Varianzen $s_x > 0$ und $s_y > 0$ und der Kovarianz s_{xy} (Sachs 1999) gegeben, dann berechnet sich der Korrelationskoeffizient *r dieser Stichprobe wie folgt:*

$$r = \frac{s_{xy}}{s_x s_y} = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{\left[\sum_{i=1}^{n} x_i - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right]\left[\sum_{i=1}^{n} y_i - \frac{1}{n}\left(\sum_{i=1}^{n} y_i\right)^2\right]}}$$

Für den Korrelationskoeffizient r gilt immer $-1 \leq r \leq 1$. Den Korrelationskoeffizient kann man als Schätzwert für den Grad der linearen Abhängigkeit von X und Y interpretieren. Die Werte 1 und -1 werden in der Praxis fast nie vorkommen, aber je näher r^2 der 1 ist, desto näher liegen die (x_i, y_i)-Punkte auf einer Geraden, d.h. desto stärker sind sie korreliert.

Wo ist aber der Zusammenhang zwischen Korrelationskoeffizient und Ähnlichkeit zweier Stichproben? Dieser Zusammenhang existiert dann, wenn wir die Elemente einer Stichprobe in eine eindeutige, mehreren Stichproben übergeordnete Reihenfolge bringen können, so dass die Tupel (x_i, y_i) zweier Stichproben X und Y eine zweidimensionale Stichprobe bilden. Der Index $i \in [1, n]$ repräsentiert dabei diese übergeordnete Reihenfolge (*Sequenz*). Solch eine übergeordnete Reihenfolge existiert z.B. beim Vergleich von gleich langen Zeichenketten (Symbolfolgen) oder bei allen zeitlich und/oder räumlich veränderlichen Größen. Bei Gleichheit beider Sequenzen müssen die Tupel (x_i, y_i), als zweidimensionale Punkte interpretiert, auf der Ursprungsgeraden liegen. Liegen die Punkte nicht auf der Ursprungsgeraden, sondern auf einer beliebigen Geraden, dann können wir immer noch von einer *linearen Ähnlichkeit* sprechen. Diese lineare Ähnlichkeit muss natürlich problemspezifisch interpretiert werden. Der Korrelationskoeffizient lässt sich somit generell zur Ähnlichkeitsanalyse von Zeitreihen einsetzen, weshalb häufig für eine Ähnlichkeitsfunktion s, im Gegensatz zu Def. 5.2.1, auch $s : M \times M \to [-1, 1]$ verlangt wird. Der Fall $r = -1$ tritt bei spiegelsymmetrischen Sequenzen auf.

5.3 Distanz

Die Distanz soll den Unterschied zwischen zwei Objekten beschreiben. Schon vom Wort her wird Distanz mit einem Zahlenwert assoziiert, d.h. die Distanz sollte umso größer sein, je größer der Unterschied zwischen beiden Objekten ist. Formal kann ein Distanzmaß wie folgt definiert werden:

Definition 5.3.1 (Distanzmaß)
Sei $M = \{m_1, \ldots, m_n\}$ eine Menge von Merkmalen, dann nennt man eine Funktion $d : M \times M \to \mathbb{R}_0^+$ Distanzfunktion, Distanzmaß, oder auch Unähnlichkeitsmaß, *wenn für $x, y \in M$ gilt:*

$$\begin{array}{lcll} (D1)\ d(x,y) & = & 0 \Leftrightarrow x = y & : \textit{ Identität} \\ (D2)\ d(x,y) & = & d(y,x) & : \textit{ Symmetrie} \end{array}$$

Distanzfunktionen die D1 erfüllen nennt man *positiv-definit*, ansonsten *positiv-semidefinit*. In der Mathematik wird im Zusammenhang mit Distanzmessungen von *Metriken* gesprochen. Metriken erfüllen im Gegensatz zu positiv-definiten Distanzfunktionen noch die sogenannte *Dreiecksungleichung*.

Definition 5.3.2 (Metrik)
Sei $d : M \times M \to \mathbb{R}_0^+$ eine Distanzfunktion (5.3.1) und gilt für $x, y, z \in M$ zusätzlich die Dreiecksungleichung

$$(D3)\ d(x,y) \leq d(x,z) + d(z,y),$$

dann nennt man d eine Metrik.

Manche Clusterverfahren verzichten jedoch auf die Erfüllung der Dreiecksungleichung (Zahn 1996). Deshalb unterscheiden wir hier zwischen Distanzfunktionen und Metriken. Im strengen Sinne wird auch die Implikation D1 von vielen Clusterverfahren nicht erfüllt, da zur Berechnung der Distanz im allgemeinen nicht alle Elemente eines Clusters herangezogen werden.

Korollar 1 (Summe von Metriken)
Seien n Metriken $d_1, \ldots, d_n$ gegeben dann ist die Summe $d_{1n}(x,y) := \sum_{i=1}^{n} d_i(x,y)$ auch eine Metrik.

Der Beweis, dass die Summe von Metriken ebenfalls die Metrikaxiome (D1-3) erfüllt, ist trivial. Die Summe positiver Werte ist positiv und somit ist $d : M \times M \to \mathbb{R}_0^+$ erfüllt. Die Summe positiver Werte kann nur dann 0 sein, wenn alle Summanden gleich 0 sind, das ist der Fall für $x = y$ und damit ist (D1) erfüllt. (D2) ist erfüllt, da die einzelnen Metriken kommutativ sind und somit auch die Summe kommutativ ist. (D3) ist erfüllt, da gilt:

$$d_{1n}(x,y) = \sum_{i=1}^{n} d_i(x,y) \leq \sum_{i=1}^{n} (d_i(x,z) + d_i(z,y)) = \sum_{i=1}^{n} d_i(x,z) + \sum_{i=1}^{n} d_i(z,y) = d_{1n}(x,z) + d_{1n}(z,y).$$

Die Verwendung von Distanzfunktionen oder Ähnlichkeitsfunktionen in Clusterverfahren ist grundsätzlich äquivalent, da Ähnlichkeit durch Distanz und Distanz durch Ähnlichkeit durch folgende Transformationsregeln ausgedrückt werden können.

Definition 5.3.3 (Transformationsregeln zwischen Ähnlichkeits- und Distanzmaß)
Sei $s : M \times M \to [0,1]$ eine Ähnlichkeitsfunktion (5.2.1) und $d : M \times M \to \mathbb{R}_0^+$ eine Distanzfunktion (5.3.1), dann kann für zwei Elemente $x, y \in M$ das Ähnlichkeitsmaß aus ihrem Distanzmaß und umgekehrt ihr Distanzmaß aus ihrem Ähnlichkeitsmaß durch die folgenden Formeln bestimmt werden:

$$\begin{aligned}
d(x,y) &:= 1 - s(x,y) \ \ \textit{mit} \ \ d(x,y) \in [0,1] \\
d(x,y) &:= \frac{1}{s(x,y)} \ \ \textit{mit} \ \ d(x,y) \in [0,\infty) \\
s(x,y) &:= 1 - \frac{d(x,y)}{\max\limits_{x,y \in M}(d(x,y))} \ \ \textit{mit} \ \ s(x,y) \in [0,1] \ \ \textit{oder} \\
s(x,y) &:= \frac{1}{d(x,y)+1} \ \ \textit{mit} \ \ s(x,y) \in [0,1)
\end{aligned}$$

5.3.1 Quantitative Distanzmaße

L_p-Metriken

Liegt ein Merkmalsraum $M \subseteq \mathbb{R}^n$ vor, so können die Distanzen mit Hilfe der allgemeinen L_p-Metriken berechnet werden, die wie folgt definiert sind:

Definition 5.3.4 (L_p-Metrik)

$$L_p(x,y) = \begin{cases} \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}, & \textit{falls } p \in [1,\infty) \\ \max\limits_{i \in [1,n]} \{|x_i - y_i|\}, & \textit{falls } p = \infty \end{cases}$$

Die L_1-Norm ist die sogenannte ***Manhattan-Distanz*** oder ***City-Block-Distanz***. Die L_2-Norm ist die klassische ***euklidische Distanz***. Den Sonderfall L_∞ bezeichnet man als ***Maximum-Distanz*** oder ***Tschebyscheff-Norm***. Es sind natürlich auch gewichtete L_p-Metriken möglich.

Quadratische Distanzform

Definition 5.3.5 (Quadratische Distanzform)
Seien $x, y \in \mathbb{R}^n$ und sei A eine $n \times n$-Ähnlichkeitsmatrix, dann lässt sich die Distanz zwischen x und y, wie folgt definieren:

$$d_A(x,y) = (x-y)A(x-y)^T \qquad \text{oder} \qquad d_A(x,y) = \sqrt{(x-y)A(x-y)^T}$$

Mahalanobis-Distanzen

Die sogenannten ***Mahalanobis-Distanzen*** sind häufig verwendete quadratische Distanzformen zum Vergleich von Clustern. Da die Mahalanobis-Distanzen auf Kovarianzmatrizen aufbauen ergeben sich nur für hinreichend große (≥ 25) Cluster sinnvolle Werte. Diese Distanzen erfüllen im allgemeinen auch nicht die Dreiecksungleichung (5.3.2) (Zahn 1996).

Definition 5.3.6 (Mahalanobis-Distanz Typ I)
Seien k Cluster $C_1, \ldots, C_k$ gegeben und bezeichnet μ_C den Mittelwertvektor, $\sum_C$ die Kovarianzmatrix und n_C die Größe eines Clusters C, dann nennt man

$$d(C_i, C_j) = (\mu_{C_i} - \mu_{C_j})^T W_{C_iC_j}^{-1}(\mu_{C_i} - \mu_{C_j}),\ \textit{mit}\ W_{C_iC_j} = \frac{1}{n_{C_i} + n_{C_j}}(n_{C_i}\textstyle\sum_{C_i} + n_{C_j}\sum_{C_j})$$

die Mahalanobis-Distanz vom Typ I.

Sind alle Nichtdiagonalelemente von $\sum_{C_i}$ und $\sum_{C_j}$ gleich Null (unabhängige Merkmale), dann ergibt sich die sogenannte ***Mittelwert-Varianz-Distanz***. Die Mahalanobis-Distanz vom Typ I ist sehr aufwendig zu berechnen, da für k Cluster, die paarweise miteinander verglichen werden sollen, $\binom{k}{2}$ Matrizen invertiert werden müssen. Kann man davon ausgehen, dass die Korrelation in allen k Clustern die gleiche ist, sollte die Mahalanobis-Distanz vom Typ II verwendet werden, da hier nur eine einzige Matrix invertiert werden muss.

Definition 5.3.7 (Mahalanobis-Distanz Typ II)
Seien k Cluster $C_1, \ldots, C_k$ gegeben und bezeichnet μ_C den Mittelwertvektor, $\sum_C$ die Kovarianzmatrix und n_C die Größe eines Clusters C, dann nennt man

$$d(C_i, C_j) = (\mu_{C_i} - \mu_{C_j})^T W_k^{-1}(\mu_{C_i} - \mu_{C_j}),\ \textit{mit}\ W_k = \frac{1}{\sum_{j=1}^k n_{C_j}} \sum_{j=1}^k n_{C_j}\textstyle\sum_{C_j}$$

die Mahalanobis-Distanz vom Typ II.

5.3.2 Nominale Distanzmaße

Hamming-Distanz

Aus der Informations- und Codierungstheorie stammt die bekannte *Hamming-Distanz* (Hamming 1987). Richard W. Hamming führte die Hamming-Distanz bei seinen Arbeiten zur Entwicklung fehlerkorrigierender Codes für die fehlerfreie Übertragung von binären Signalen ein. Der verallgemeinerte M-Koeffizient (Def. 5.2.5) ist eine direkte Ableitung aus der Hamming-Distanz.

Definition 5.3.8 (Hamming-Distanz)

$$d(x,y) = \sum_{i=1}^{n} h(x_i, y_i) \quad mit \quad h(x_i, y_i) = \begin{cases} 1, & falls\ x_i \neq y_i \\ 0 & sonst \end{cases}$$

Levenshtein-Distanz

Im Falle beliebig langer, aber nicht notwendig gleich langer Symbolfolgen bietet sich die sogenannte *Levenshtein-Distanz* (Levenshtein 1965) an. Die Levenshtein-Distanz ist nach dem russischen Wissenschaftler Vladimir Levenshtein benannt, der diesen Algorithmus im Jahre 1965 veröffentlichte. Die Levenshtein-Distanz wird auch häufig mit *Edit-Distanz* bezeichnet.

Definition 5.3.9 (Levenshtein-Distanz)
Die Levenshtein-Distanz *zweier Symbolfolgen (Strings) x und y ist die* ***minimale*** *Anzahl von* Editieroperationen, *um den String x in den String y zu transformieren, wenn folgende Editieroperationen zugelassen sind:*

Einfügen, ***Löschen*** *und* ***Ersetzen***. *Die* Levenshtein-Distanz *lässt sich wie folgt rekursiv definieren:*

$$d(x,y) = d(x_{1,n}, y_{1,m}) mit$$

$$d(x_{1,i}, y_{1,j}) = \begin{cases} j, & falls\ i = 0 \\ i, & falls\ j = 0 \\ d(x_{1,i-1}, y_{1,j-1}), & falls\ i,j > 0\ und\ x_i = y_j \\ \min\{d(x_{1,i-1}, y_{1,j-1}) + 1, d(x_{1,i-1}, y_{1,j}) + 1, d(x_{1,i}, y_{1,j-1}) + 1\} & sonst \end{cases}$$

Zur Berechnung der Levenshtein-Distanz wird die Methode der *Dynamischen Programmierung* (Bellman 1957, Sniedovich 1992) verwendet. Die Levenshtein-Distanz zwischen zwei Strings mit den Längen m und n kann im allgemeinen in $O(mn)$ berechnet werden (Levenshtein 1965, Sankoff & Kruskal 1983, Ukkonen 1985, Arslan & Egecioglu 2000).

Die Levenshtein-Distanz wird in Bereichen, wie z.B. dem Text Mining (Data Mining auf Texten), der automatischen Rechtschreibprüfung, der maschinellen Spracherkennung, der computergestützten DNA-Analyse und der sogenannten Plagiat-Erkennung (literarischer Diebstahl, Nachahmung) angewendet.

Die Hamming-Distanz kann als ein Sonderfall der Levenshtein-Distanz angesehen werden. Erlaubt man als einzige Editieroperation die Operation *Ersetzen* und vergleicht nur gleich lange Strings, dann berechnet man somit die Hamming-Distanz.

Die Levenshtein-Distanz kann auch in dem Sinne verallgemeinert werden, dass man durch eine Gewichts- oder Kostenfunktion jeder Editieroperation ein Gewicht (Kosten) zuordnet und nicht nach der minimalen Anzahl von Editieroperationen sucht, sondern nach der Folge von Editieroperationen mit dem geringsten Gewicht (den geringsten Kosten). Das minimale Gewicht (der minimale Kostenaufwand) ist dann die Distanz zwischen beiden Symbolfolgen.

Ergänzt man die Editieroperationen noch um die Operation *Vertausche zwei benachbarte Symbole*, dann erhält man die sogenannte Damerau-Levenshtein-Distanz (Damerau 1964, Pfeifer, Poersch & Fuhr 1995).

5.4 Distanz- und Ähnlichkeitsmaße basierend auf Hintergrundwissen

In den vorigen Abschnitten haben wir einige Maße zum Vergleich verschiedener Datentypen aufgezählt. Im allgemeinen bestehen die Daten (Objekte) eines betrachteten Datensatzes oder einer Datenbank aus Merkmalen unterschiedlicher Datentypen. Um solche komplexen Daten miteinander zu vergleichen, kann man im einfachsten Fall die Werte aller vorkommenden Merkmale als nominale Daten auffassen und dann z.B. die Levenshtein- oder

Hamming-Distanz anwenden. Dieses Vorgehen wird in den meisten Fällen keine sinnvollen Ergebnisse erbringen. Nehmen wir als Beispiel einen Datensatz aus dem *Amtlichen Liegenschaftskataster-Informationssystem* ***ALKIS***, in dem geometrische Informationen und Sachinformationen simultan gespeichert sind. Betrachten wir nun jedes Objekt dieses Datensatzes, der Informationen über die Geometrieart (Punkt, Linie, Fläche), Lage (Koordinaten) und Nutzungsarten enthält, als eine einzige große Symbolfolge, dann ist es höchst unwahrscheinlich, dass wir durch ein Clusterverfahren eine Gruppierung erhalten, die die räumliche Nähe, die Form und die Nutzung der Objekte *sinnvoll* berücksichtigt. Das Verfahren wird einzig eine Zerlegung liefern, die die Ähnlichkeit dieser Symbolfolgen berücksichtigt. Um eine *sinnvolle* Gruppierung zu erhalten, muss also immer spezifisches Hintergrundwissen, also ein Modell, zugrunde gelegt werden. Das einfachste Modell kann dadurch gebildet werden, dass man alle vorkommenden Merkmale eines Objektes definiert und zu jedem Merkmal den Skalentyp und die zu verwendende Distanz- oder Ähnlichkeitsfunktion vorgibt. Wird nun jedes Objekt als Merkmalsvektor betrachtet, dann kann für diesen Vektor unter Berücksichtigung von Korollar 1 (Seite 43) und Definition 5.3.3 eine Distanz- oder Ähnlichkeitsfunktion definiert werden.

Diese Definition ist jedoch im Falle unseres Beispiels immer noch nicht ausreichend, wenn wir eine vernünftige Gruppierung bzgl. der Nutzungsart erwarten. In der ALK existieren für Gebäude 87 verschiedene Nutzungsarten, die in Form einer Zahl (Gebäudeschlüssel) oder als Zeichenkette zur Verfügung stehen. Als Beispiel sollen die folgenden Gebäudenutzungen dienen, dabei ist der erste Wert der Gebäudeschlüssel, der zweite die Textform und der dritte Wert die Bedeutung in der Realität:

- 2366 : Gar → Garage
- 2361 : Phs → Parkhaus
- 2359 : Aufzug → Aufzug
- 1411 : Büro → Bürogebäude
- 1409 : Ghs → Geschäftshaus
- 1301 : Whs → Wohnhaus
- 1841 : Wghs → Wohn- und Geschäftshaus
- 1842 : Wbüro → Wohn- und Bürogebäude

Wenn wir nun als *Mensch* solche Gebäudeobjekte miteinander vergleichen – unter der Annahme das wir nach Objekten möglichst gleicher Nutzung suchen – dann werden wir z.B. feststellen, dass Garage und Parkhaus ähnlich sind. Einen Aufzug und eine Garage würden wir jedoch als völlig verschieden einstufen. Basierend auf den oben definierten Zahlenwerten und Zeichenketten wird man jedoch keinerlei vernünftiges Ergebnis erzielen können. Die Levenshtein-Distanz zwischen *Gar* und *Phs* wäre 3, dagegen wäre die Levenshtein-Distanz zwischen *Gar* und *Aufzug* mit 6 größer. Dieses Ergebnis sieht auf den ersten Blick vernünftig aus, beruht jedoch nur darauf, dass Aufzug nicht abgekürzt wurde. Hätte man für Aufzug zum Beispiel *Auf* verwendet, wäre die Distanz ebenfalls 3. Wenn wir die Zahlenwerte betrachten und den Betrag der Differenz als Abstandsmaß verwenden, erhalten wir als Distanz zwischen Garage und Parkhaus 5 sowie zwischen Garage und Aufzug 7. Jedoch wäre der Abstand zwischen Parkhaus und Aufzug 2 und somit ein Aufzug und ein Parkhaus erheblich ähnlicher, als ein Parkhaus und eine Garage. Ein Parkhaus besitzt meistens einen Aufzug, jedoch ist ein Parkhaus kein Aufzug und somit ist diese Distanz sinnlos. Ähnlich argumentieren kann man für die Schlüsselwerte und Zeichenketten von Bürogebäude, Geschäftshaus, Wohnhaus und deren Kombinationen.

Wie man sieht genügt es nicht, einfach nur eine dem Skalentyp angepasste Ähnlichkeits- oder Distanzfunktion zu wählen, sondern man muss auch berücksichtigen, ob der Skalentyp ein Merkmal direkt repräsentiert. Ist das betrachtete Merkmal z.B. die Körpergröße in cm, dann ist ein Zahlentyp oder ein ordinaler Skalentyp eine angemessene Darstellung, da das Merkmal *Körpergröße* eine messbare, physikalische Eigenschaft ist. Ist das Merkmal andererseits ein nominaler Skalentyp, wie z.B. der Name oder die Adresse einer Person oder ein Ausschnitt der DNA eines Lebewesen, dann muss man unterscheiden, ob diese Symbolfolgen oder die „Bedeutungen“, die hinter diesen nominalen Typen stehen, von Interesse sind. Anders ausgedrückt: sind wir nur am Namen einer Person oder der Person selbst interessiert? Genauso ist es mit der Nutzung von Gebäuden. Jegliche Art der Bezeichnung einer *Gebäudenutzung* kann nur eine Codierung der komplexen *Bedeutung*, die hinter dem Begriff Gebäudenutzung steht, darstellen. Im allgemeinen wird man immer eine Art von Hintergrundwissen (Modell), das mehr oder weniger komplex sein kann, benötigen und einem Clusteralgorithmus in geeigneter Form zur Verfügung stellen.

In vielen Fällen kann menschliches Wissen in hierarchischer Form geordnet werden. Eine solche hierarchische Ordnung von Begriffen oder Objekten (Individuen), die wir im weiteren mit der Bedeutung, die hinter diesen

Begriffen steht, gleichsetzen wollen, nennt man auch *Taxonomie*. Es gibt zum Beispiel die Taxonomie der Giftplanzen oder die Taxonomie der wirbellosen Tiere. Eine solche Taxonomie kann in einer *Konzepthierarchie* repräsentiert werden. Eine Konzepthierarchie ist im *objektorientierten* Sinne eine *Vererbungshierarchie*, in der im allgemeinen auch *Mehrfachvererbung* zugelassen ist. Einige formale Ansätze modellieren Konzepthierarchien als *Verbände* (Hermes 1967). Dies ist immer dann möglich und angemessen, wenn alle möglichen Merkmale oder Ausprägungen der Objekte einer Konzepthierarchie bekannt sind, da man dann durch einen *Teilmengenverband* über der Menge dieser Merkmale eine Konzepthierarchie erzeugen kann. Dieser Ansatz funktioniert jedoch nicht, wenn man es mit Konzepten zu tun hat, die einfach nur durch ihren Bezeichner (Namen) repräsentiert werden und nicht durch ihre Merkmalsausprägung. Das ist immer der Fall, wenn die Merkmalsausprägung nicht bekannt ist oder implizit angenommen wird und deshalb nicht im Datensatz oder der Datenbank gespeichert wird. In einem Teilmengenverband kann ein solches Konzept nur als leere Menge dargestellt werden, was jedoch das *kleinste Objekt* in einem Teilmengenverband darstellt und *kein Konzept* bedeutet. Als Beispiel hierfür soll wieder die Gebäude- oder Flächennutzung in der ALK dienen, in der die Nutzungsarten nicht durch Merkmale beschrieben werden, sondern als fest definiert (dem Nutzer bekannt) angenommen werden.

Wir definieren als Konzepthierarchie (siehe Abb. 5.2) also eine Vererbungshierarchie (endlicher azyklischer gerichteter Graph) mit den folgenden Eigenschaften:

- Die Knoten dieses Graphen[2] repräsentieren die bekannten Konzepte und damit das vorhandene *Wissen*. Ein Konzept ist eine Klasse von Objekten (Instanzen), die eine logisch zusammengehörende Menge von Werten darstellt oder als eine allgemein bekannte Bedeutung zu verstehen ist.
- Der Graph besitzt einen besonders ausgezeichneten Knoten, die *Wurzel*, den wir als *leeres Konzept* ($\emptyset$) bezeichnen wollen.
- Die gerichteten Kanten des Graphen beschreiben eine *Ist-ein-Beziehung*. Das Konzept am Ende der gerichteten Kante nennt man *Überkonzept*. Die Klasse am Anfang der gerichteten Kante nennt man *Unterkonzept*.
- Jedes Konzept kann eine Menge von Merkmalen besitzen. Jedes Merkmal besitzt einen Namen als Bezeichner und ist einem Konzept der Hierarchie zugeordnet. Jedes Merkmal ist durch Name und zugehöriges Konzept eindeutig definiert und kann in einem Konzept nur einmal vorkommen.
- Ein Unterkonzept *erbt* alle Merkmale seiner Überkonzepte und kann noch weitere Merkmale definieren. Jedoch kann kein Merkmal *redefiniert* werden, d.h. existiert ein Merkmal $x = (id_x : C_x)$ in einem Überkonzept $\mathcal{O}$, dann wird durch die Definition $x = (id_x : C_x)$ in einem Unterkonzept $\mathcal{U}$ von $\mathcal{O}$ kein neues Merkmal in $\mathcal{U}$ eingeführt.
- Konzepte, die direkte Unterkonzepte des leeren Konzepts sind, nennt man *atomare Konzepte* oder *Axiome*, da sie von *nichts* abgeleitet wurden und als gegeben definiert sind. Man kann solche Konzepte auch als *Grundwissen* bezeichnen. Entspricht ein atomares Konzept einem Skalentyp, so nennen wir es ein *Skalenkonzept*.

5.4.1 Distanzmaß auf Konzepthierarchien

Im folgenden wollen wir nun eine Metrik definieren, um die Distanz zwischen zwei Konzepten bestimmen zu können und darauf aufbauend werden wir eine Distanzfunktion definieren, die den Unterschied zwischen Instanzen von Konzepten (Objekten) bestimmt. Ein Distanzmaß für Konzepte einer gegebenen Konzepthierarchie sollte folgende Eigenschaften aufweisen:

K1: Die Ähnlichkeit (Distanz) eines Konzepts mit sich selber ist 1 (0).

K2: Die Ähnlichkeit (Distanz) zweier unterschiedlicher Konzepte ist 0 (∞), wenn sie kein gemeinsames Überkonzept besitzen, d.h. das einzige gemeinsame Überkonzept ist das leere Konzept.

[2]Die Definition graphentheoretischer Begriffe, wie z.B. gerichtete Kante, Weg und Länge eines Weges findet man im Abschnitt 6.1 des folgenden Kapitels über Nachbarschaftsgraphen.

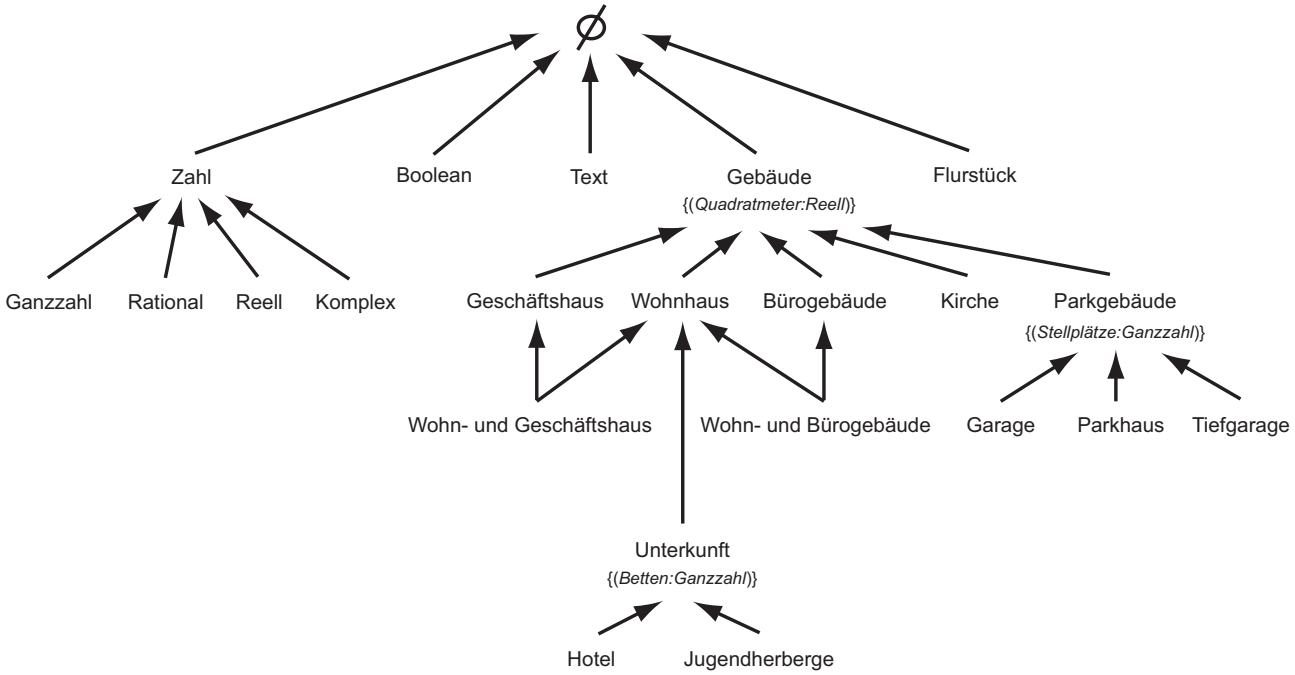


Abbildung 5.2: Beispiel für eine Konzepthierarchie

K3: Die Ähnlichkeit (Distanz) zweier unterschiedlicher Konzepte ist ungleich 1 (0), selbst wenn sie die exakt gleichen Merkmale besitzen.

K4: Die Ähnlichkeit (Distanz) zweier unterschiedlicher Konzepte ist umso kleiner (größer), je mehr unterschiedliche Merkmale beide Konzepte besitzen.

K1 wird verlangt, um die Symmetrieeigenschaft von Ähnlichkeits- und Distanzmaßen zu erfüllen. K2 drückt aus, dass Konzepte ohne gemeinsames Überkonzept als völlig verschieden aufzufassen sind. K3 verlangt, dass jedes Konzept unabhängig von seiner Merkmalsausprägung einen eindeutig von den anderen Konzepten unterscheidbaren Begriff repräsentiert und somit in einer Konzepthierarchie keine zwei Konzepte mit exakt der gleichen Bedeutung existieren können. Mit K4 wird verdeutlicht, dass nicht nur der Abstand zum gemeinsamen Überkonzept ausschlaggebend für die Ähnlichkeit (Distanz) ist, sondern dass sich auch die Merkmalsausprägung von Konzepten wesentlich auf die Ähnlichkeit (Distanz) auswirkt. Das ist besonders wichtig zur Unterscheidung von Konzepten gleicher Hierarchiestufe (z.B. alle direkten Unterkonzepte eines Überkonzepts).

Eine, wie oben definierte, Konzepthierarchie kann nun dazu genutzt werden, um auch Objekte miteinander zu vergleichen, die Instanzen eines Konzepts sind, das keinem Skalentyp entspricht. Wir gehen im weiteren davon aus, dass für jedes angegebene Skalenkonzept eine geeignete Distanzfunktion definiert ist, um Instanzen dieses Konzepts miteinander vergleichen zu können. Unter diesen Voraussetzungen kann man nun die Distanz zwischen zwei Konzepten wie folgt definieren:

Definition 5.4.1 (Konzept-Distanz)
Seien C_i und C_j Konzepte der gleichen Konzepthierarchie $\mathcal{K}$. Mit $\delta_{\mathcal{K}}(C_i, C_j)$ bezeichne man die Länge des kürzesten Weges von C_i nach C_j in $\mathcal{K}$, wobei $\delta_{\mathcal{K}}(C_i, C_j) = \infty$ gilt, falls der kürzeste Weg das leere Konzept enthält ($\emptyset$) und bezeichnet $\mathcal{M}(C_i)$ die Merkmalsmenge von Konzept C_i, sowie $|\mathcal{M}(C_i)|$ die Anzahl der Merkmale, dann sei

$$\Delta_{\mathcal{K}}(C_i, C_j) = \delta_{\mathcal{K}}(C_i, C_j) + |(\mathcal{M}(C_i) \cup \mathcal{M}(C_j)) \setminus (\mathcal{M}(C_i) \cap \mathcal{M}(C_j))|$$

die Konzept-Distanz *zwischen den beiden Konzepten C_i und C_j in $\mathcal{K}$.*

Anhand von einigen Beispielen soll die Konzept-Distanz verdeutlicht werden. Wenn wir von der Konzepthierarchie aus Abbildung 5.2 ausgehen, kann man z.B. die folgenden Konzept-Distanzen berechnen:

- $\Delta_{\mathcal{K}}$(Gebäude,Text) $= \infty$, da ihr gemeinsames Überkonzept $\emptyset$ ist.
- $\Delta_{\mathcal{K}}$(Wohnhaus,Wohn- und Geschäftshaus) $= 1$, da das Konzept Wohnhaus das direkte Überkonzept vom Konzept Wohn- und Geschäftshaus ist und keine unterschiedlichen Merkmale existieren. Gleiches gilt für die Distanz
- $\Delta_{\mathcal{K}}$(Geschäftshaus,Wohn- und Geschäftshaus) $= 1$.
- $\Delta_{\mathcal{K}}$(Bürogebäude,Wohn- und Geschäftshaus) $= 3$, da zwei Wege mit der Länge 3 existieren und beide Konzepte die gleichen Merkmale besitzen.
- $\Delta_{\mathcal{K}}$(Wohn- und Bürogebäude,Wohn- und Geschäftshaus) $= 2$, da beide Konzepte die gleichen Merkmale besitzen und von den drei möglichen Wegen zwischen beiden Konzepten der kürzeste Weg die Länge 2 hat.
- $\Delta_{\mathcal{K}}$(Hotel,Parkhaus) $= 7$, da die Länge des kürzesten Weges zwischen beiden Konzepten 5 beträgt und es 2 unterschiedliche Merkmale $(\{(Betten : Ganzzahl), (Stellplätze : Ganzzahl)\})$ gibt.

Mit Hilfe der Konzept-Distanz und der Summenmetrik (Seite 43, Korollar 1) können wir nun die Distanz zwischen Instanzen von Konzepten wie folgt definieren:

Definition 5.4.2 (Objekt-Distanz)
Seien die Objekte O_i^p und O_j^q Instanzen der Konzepte C_p und C_q der Konzepthierarchie $\mathcal{K}$ und seien für alle Skalenkonzepte *geeignete Distanzfunktionen δ_r definiert und bezeichne man mit $m_{i,r}^p$ die Wertbelegung des Merkmals $M_{i,r}^p$ von Objekt O_i^p, dann sei*

$$d_{\mathcal{K}}(O_i^p, O_j^q) = \Delta_{\mathcal{K}}(C_p, C_q) + \sum_{\{r|M_{i,r}^p = M_{j,r}^q\}} \delta_r(m_{i,r}^p, m_{j,r}^q)$$

die Objekt-Distanz *zwischen den beiden Objekten O_i^p und O_j^q.*

Nehmen wir nun an, wir hätten bzgl. der Konzepthierarchie aus Abbildung 5.2 die folgenden drei Konzeptinstanzen (Objekte):

- Hotel $\mathcal{H}_1 = \{(Quadratmeter : 340, 0), (Betten : 100)\}$,
- Hotel $\mathcal{H}_2 = \{(Quadratmeter : 510, 75), (Betten : 350)\}$ und
- Parkhaus $\mathcal{P} = \{(Quadratmeter : 340, 0), (Stellplätze : 100)\}$.

Dann ergeben sich die folgenden Objekt-Distanzen, wenn wir für die quantitativen Skalentypen Reell und Ganzzahl die Metrik $\delta(x, y) = |x - y|$ verwenden:

- $d_{\mathcal{K}}(\mathcal{H}_1, \mathcal{H}_2) = 0 + |340,0 - 510,75| + |100 - 350| = 0 + 170,75 + 250 = 420,75$
- $d_{\mathcal{K}}(\mathcal{H}_1, \mathcal{P}) = 7 + |340,0 - 340,0| = 7 + 0 = 7$
- $d_{\mathcal{K}}(\mathcal{P}, \mathcal{H}_2) = 7 + |340,0 - 510,75| = 7 + 170,75 = 177,75$

Die Objekt-Distanz ist also nichts weiteres als die Addition der Summe der Distanzen von Wertebelegungen aller gleichen Merkmale von zwei Objekten zu ihrer Konzept-Distanz. Die Konzept- und Objekt-Distanz erfüllen die Bedingungen D1 bis D3, da die absolute Differenz zweier Werte und die Weglänge in einem Graphen Metriken sind und die Summe von Metriken ebenfalls eine Metrik ist (Seite 43, Korollar 1).

Wir haben nun gezeigt, wie man mit Hilfe von Konzepthierarchien als Hintergrundwissen ein geeignetes Distanzmaß definieren kann. Wie kommt man nun an eine geeignete Konzepthierarchie? Im allgemeinen muss die Konzepthierarchie durch den Nutzer vorgegeben werden, jedoch werden auch automatische Verfahren zur Ableitung von Konzepthierarchien aus Datenbanken untersucht. In letzter Zeit verstärken sich auch im GIS-Bereich

die Arbeiten zur Definition von allgemeinen Konzepthierarchien für geographische Problemdomänen. Man findet Literatur hierzu meistens unter dem Schlagwort *Ontologie* oder *Ontology*. Aus streng philosophischer Sicht ist unseres Erachtens diese Bezeichnung jedoch nicht korrekt, denn die Ontologie ist die Wissenschaft (Theorie, Untersuchung) *des Seins* bzw. die Erforschung dessen, *was ist, wie es ist*; also weit mehr als die Findung und Einordnung von Begriffen für *Seiendes*. Das Thema Ontologie auf dem Gebiet der Geoinformatik umfasst im wesentlichen die automatische Ableitung von Konzepthierarchien oder Taxonomien (Kuhn 2001). Im Englischen unterscheiden deshalb auch manche Autoren zwischen *Ontology* für die philosophische Interpretation von Ontologie und für die Interpretation von Ontologie in der maschinellen Wissensverarbeitung *Ontologies*. Eine gute Einführung in das Thema *Ontologie im GIS-Bereich* findet man in (Winter 2001).

5.5 Distanzmaße für Objektmengen

Bisher haben wir Maße beschrieben, um einzelne Objekte, die beliebig viele Merkmale besitzen können, miteinander vergleichen zu können. Clusterverfahren müssen im allgemeinen jedoch nicht nur einzelne Objekte miteinander vergleichen, sondern auch Mengen von Objekten. Hierzu bieten sich die im vorigen Kapitel 4 erwähnten klassischen Cluster-Distanzmaße an, die hier noch einmal aufgezählt sind:

Definition 5.5.1 (Cluster-Distanzen)
Single-Linkage-Verfahren: $d(I,J) := \min\limits_{x\in I, y\in J} d(x,y)$

Complete-Linkage-Verfahren: $d(I,J) := \max\limits_{x\in I, y\in J} d(x,y)$

Average-Linkage-Verfahren: $d(I,J) := \frac{1}{|I||J|} \sum\limits_{x\in I, y\in J} d(x,y)$

Ward-Verfahren: $d(I,J) := \frac{2|I||J|}{|I|+|J|} d(\mu_I, \mu_J)$

Diese Maße bauen direkt auf den oben beschriebenen Distanzmaßen zwischen einzelnen Objekten auf, um die Distanz zwischen zwei Clustern zu berechnen. Das Single-Linkage-Verfahren verwendet den kleinsten Abstand zwischen den Objekten zweier Clusters. Das Complete-Linkage-Verfahren verwendet dagegen den größten Abstand und das Average-Linkage-Verfahren den mittleren Abstand. Clusterverfahren, die jeden Cluster durch einen Repräsentanten (Medoid) darstellen, verwenden direkt Objekt-Distanzmaße und vergleichen mit diesen Maßen jeweils nur die Repräsentanten der Cluster und nicht alle Objekte der Cluster. Es sind natürlich auch andere Definitionen möglich, wie z.B. gewichtete Varianten der genannten Cluster-Distanzen. Ein weiteres bekanntes Maß zur Bestimmung des Abstands zweier Mengen ist die sogenannte *Hausdorff-Metrik*.

Definition 5.5.2 (Hausdorff-Metrik)
Sei $\mathbb{M}$ ein metrischer Raum mit der Metrik δ und $X, Y \subset \mathbb{M}$ kompakt, dann definiert sich die Hausdorff-Metrik *$h(X,Y)$ wie folgt:*

$$\begin{aligned} h(X,Y) &= \max\{d(X,Y), d(Y,X)\} \text{ mit} \\ d(X,Y) &= \max\{d(x,Y) \mid x \in X\} \text{ und} \\ d(x,Y) &= \min\{\delta(x,y) \mid y \in Y\} \end{aligned}$$

5.6 Diskussion

In diesem Kapitel haben wir verschiedene Möglichkeiten der Ähnlichkeits- und Distanzbestimmung diskutiert und aufgezeigt, dass kein eindeutiges Maß existiert, sondern dass man den *Skalentyp* und die *Bedeutung* eines Wertes berücksichtigen muss und deshalb eine sinnvolle Ähnlichkeitsbestimmung ohne Hintergrundwissen nicht möglich ist. Die beschriebenen Ähnlichkeitsmaße zwischen Objekten oder Mengen berücksichtigen jedoch immer nur zwei Objekte oder zwei Mengen, um ein Maß für die Ähnlichkeit (Abstand) zu bestimmen und wollen

dies als *direkte Ähnlichkeit* bezeichnen. Die direkte Ähnlichkeit berücksichtigt jedoch nicht die *Nachbarschaft* (Umgebung) dieser Objekte oder Mengen. Im Gegensatz dazu bezieht ein sogenanntes *relatives Ähnlichkeitsmaß* die Nachbarschaft der betrachteten Objekte (Mengen) mit ein und wird den beiden Objekten nur dann eine direkte Ähnlichkeit (Distanz) ungleich 0 (∞) zuordnen, wenn die direkte Ähnlichkeit beider Objekte (Mengen) in einer vorgegebenen Relation zu den direkten Ähnlichkeiten (Distanzen) zwischen beiden Objekten (Mengen) und den anderen Objekten (Mengen) steht. Die Ähnlichkeitsmatrizen relativer Maße sind erheblich dünner besetzt als die Ähnlichkeitsmatrizen direkter Maße, was bei Clusterverfahren zu einer Einschränkung des Suchraums führt. Im folgenden Kapitel werden wir nun auf die sogenannten *Nachbarschaftsgraphen* eingehen, die uns solche relativen Ähnlichkeitsmaße zur Verfügung stellen.

Kapitel 6

Nachbarschaftsgraphen

Im vorherigen Kapitel sind wir auf das grundlegende Problem der Definition von Ähnlichkeit zwischen Objekten aus beliebigen Mengen eingegangen und haben dabei die gleichwertige Interpretation von Nachbarschaft und relativer Ähnlichkeit beschrieben. Da wir im weiteren raumbezogene Daten gruppieren wollen, stellt sich nun die Frage nach einer geeigneten Definition für Nachbarschaft. Man kann die sogenannte *4er-* oder *8er-Nachbarschaft* aus der Rasterdatenverarbeitung verwenden, in dem man ein Zellraster so bestimmt, dass maximal ein Objekt in jeder Zelle enthalten ist (Abb. 6.1 a)). Eine andere Möglichkeit der Definition von Nachbarschaft kann durch Vorgabe eines festen Abstands erfolgen, wie in Abbildung 6.1 b) dargestellt oder man verwendet das sogenannte *Voronoi-Diagramm* (Abb. 6.1 c)) zur Definition der Nachbarschaft. Wir werden im folgenden zur Definition von Nachbarschaft die sogenannten Nachbarschaftsgraphen verwenden. Abbildung 6.1 d) zeigt die *Delaunay-Triangulation* einer Punktmenge, die der bekannteste Typ von Nachbarschaftsgraph neben dem Nächster-Nachbar-Graph ist und der duale Graph des Voronoi-Diagramms ist. Nachbarschaftsgraphen stellen in dem von uns entwickelten parameterfreien Clusterverfahren, das in Kapitel 7 genau beschrieben wird, das Fundament zur Definition von Ähnlichkeit dar und sollen deshalb im folgenden genauer beschrieben werden.

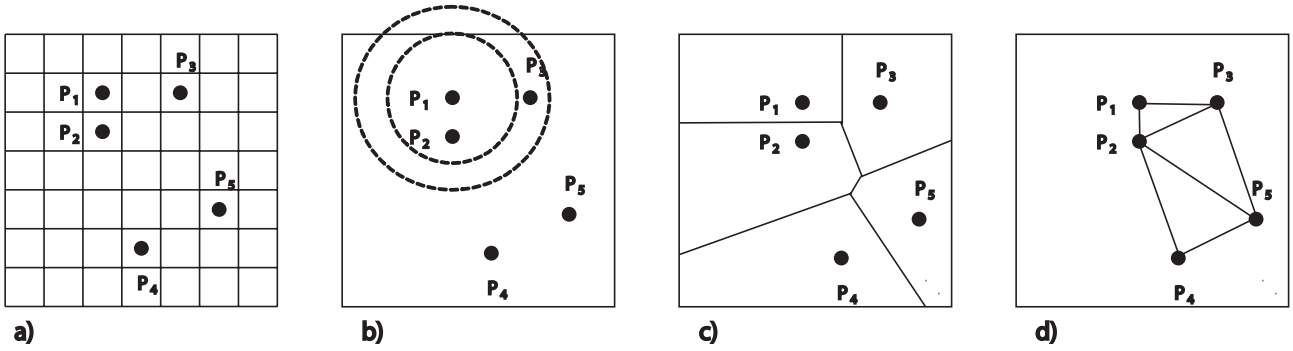


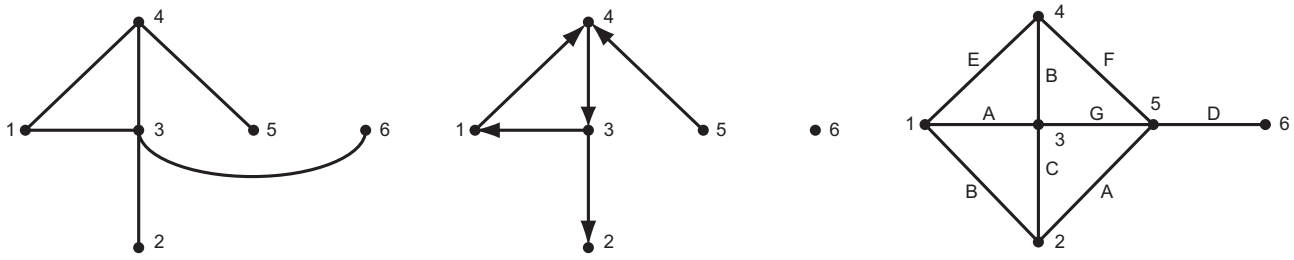
Abbildung 6.1: Verschiedene Arten der Nachbarschaft (adaptiert nach (Estivill-Castro & Lee 2002)): a) Raster, b) fester Abstand, c) Voronoi-Diagramm, d) Delaunay-Triangulation.

6.1 Graphen

Da Nachbarschaftsgraphen eine spezielle Teilmenge der Menge aller Graphen sind, wollen wir zuerst auf die wesentlichen Begriffe aus der Graphentheorie eingehen. Diese Arbeit entstand aufgrund von Problemen in der Automatisierung von Aufgaben in Geo-Informationssystemen und der Kartographie, deshalb ist es interessant anzumerken, dass bei der Entwicklung der Graphentheorie, die noch ein relativ junges Gebiet der Mathematik darstellt, das *4-Farben Problem*[1] eine wesentliche Rolle spielte. Heutzutage findet die Graphentheorie ein breites

[1] Kann jede ebene Landkarte stets mit höchstens 4 Farben so eingefärbt werden, dass keine zwei angrenzenden Länder die gleiche Farbe haben? Diese Frage stellte Frederick Guthrie 1852 August de Morgan und es dauerte 124 Jahre, diese Frage mit Ja zu beantworten.

Anwendungsfeld. Beispielhaft seien genannt: Informatik (Compilerbau, Betriebssysteme, Computernetzwerke, Künstliche Intelligenz), Chemie, Physik und die Anwendungsgebiete des Operations Research.[2] Zur allgemeinen Einführung in die Graphentheorie und für weiterführende Literatur verweisen wir z.B. auf (Diestel 2000, Aigner 1984, Hein 1977, Kaufmann 1971, Sedlacek 1971, Reinhardt & Soeder 1984).



(a) ungerichtet (b) gerichtet (c) bewertet

Abbildung 6.2: Graphen

6.1.1 Definitionen von Graphen

Definition 6.1.1 (Graph)
Das Tripel $\mathbf{G} := (\mathbf{E}, \mathbf{K}, \mathcal{I})$ *heißt* Graph *mit den disjunkten Mengen* $\mathbf{E}(\mathbf{G})$ *(*Eckenmenge*) und* $\mathbf{K}(\mathbf{G})$ *(*Kantenmenge*)*[3] *und der Abbildung* $\mathcal{I}$ *zur Beschreibung der* Inzidenzbeziehungen *von Ecken und Kanten*[4].

Die Inzidenzbeziehungen lassen sich durch geordnete Paare darstellen. Seien x *und* y *zwei Ecken und* k *die Kante mit der* x *und* y *inzidieren (*x *und* y *sind durch* k *miteinander verbunden), dann können wir der Kante* k *das Tupel* (x, y) *oder* (y, x) *zuordnen und nennen* x *und* y *die* Endecken *von* k*. Anstelle von* (x, y) *schreibt man auch kurz* xy*.*

Man nennt einen Graphen **ungerichtet** *(Abb. 6.2a)), wenn keine der beiden Endecken einer Kante besonders ausgezeichnet sind, d.h. es gilt die Äquivalenzrelation* $\ddot{A} : xy = yx$*. Die Menge* $E \times E \setminus \ddot{A}$ *besteht dann aus Klassen der Form* $[[xx]] = \{xx\}$ *oder* $[[xy]] = \{xy, yx\}$*, falls* $x \neq y$ *ist. Es gilt also* $\mathcal{I} : \mathbf{K} \to \mathbf{E} \times \mathbf{E} \setminus \ddot{\mathbf{A}}$*.*

Wird eine Ecke einer Kante als Startecke *ausgezeichnet, nennt man den Graphen* **gerichtet** *(Abb. 6.2b)) und die Äquivalenzrelation* $\ddot{A} : xy = yx$ *entfällt. Es gilt dann* $\mathcal{I} : \mathbf{K} \to \mathbf{E} \times \mathbf{E}$*. Jede Kante* $xy \in G$ *besitzt dann die Richtung* **von** x **nach** y*. Ein gerichteter Graph wird auch als* Digraph *bezeichnet.*

Zwei Ecken x, y *von* G *sind* benachbart *oder* adjazent *und heißen* Nachbarn *voneinander, wenn* $xy \in K(G)$ *gilt. Zwei Kanten* $k \neq l$ *heißen benachbart, wenn sie eine gemeinsame Endecke besitzen. Paarweise nicht benachbarte Kanten und Ecken von* G *heißen auch* unabhängig*.*

Definition 6.1.2 (Komplementärer Graph)
Das Komplement $\overline{\mathbf{G}}$ *eines Graphen* G *ist der Graph auf* $E(G)$*, in dem zwei Ecken genau dann benachbart sind, wenn sie es nicht in* G *sind. Formal geschrieben:*

$$\overline{\mathbf{G}} := (\mathbf{E}, \overline{\mathbf{K}}) \textbf{ mit } \overline{\mathbf{K}} = (\mathbf{E} \times \mathbf{E}) \setminus \mathbf{K}$$

[2] Operations Research ist ein auf praktische Anwendung mathematischer Methoden ausgerichteter Wissenszweig und befasst sich mit der Problemanalyse und Vorbereitung optimaler Entscheidungen in Organisationen. Operations Research ist geprägt durch die Zusammenarbeit von Mathematik, Wirtschaftswissenschaften und Informatik, siehe auch http://www.gor-ev.de/.

[3] Es sei angemerkt, dass in der internationalen Literatur die Schreibweise $G = (V, E)$ verwendet wird. E steht dabei für *Edge Set* und V für *Vertex Set*.

[4] Die Inzidenzabbildung wird im allgemeinen als implizit gegeben angenommen und die kurze Schreibweise $G = (E, K)$ bevorzugt.

Definition 6.1.3 (Eckengrad)
Sei $x \in E(G)$, dann nennt man die Anzahl aller Kanten $|K(x)|$, die mit x inzidieren, den Grad von x *(oder die* Valenz*) und schreibt dafür $grad_G(x) = grad(x)$ oder häufiger abgeleitet vom englischen Wort* Degree $d_G(x) = d(x)$*. Eine Ecke mit dem Grad 0 bezeichnet man als* isoliert*.*

In gerichteten Graphen unterscheidet man zwischen dem **Eingangsgrad** $grad^+_G(x) = grad^+(x)$ *und dem* **Ausgangsgrad** $grad^-_G(x) = grad^-(x)$ *einer Ecke x. Der Eingangsgrad ist die Anzahl aller Kanten $|E^+(x)|$, die mit x inzidieren und x ist nicht Startecke dieser Kanten. Der Ausgangsgrad ist die Anzahl aller Kanten $|E^-(x)|$, die mit x inzidieren und x ist Startecke dieser Kanten. In gerichteten Graphen gilt für den Grad einer Ecke x: $grad(x) = grad^+(x) + grad^-(x)$.*

Minimalgrad *von G heißt die Zahl $\delta(G) := \min\{grad(x) \mid x \in E(G)\}$ und $\Delta(G) := \max\{grad(x) \mid x \in E(G)\}$ heißt* **Maximalgrad**. *Der* **Durchschnittsgrad** *von G ist definiert durch die Zahl*

$$grad(G) := \sum_{x \in E(G)} grad(x)/|E(G)|$$

und es gilt offenbar

$$\delta(G) \leq grad(G) \leq \Delta(G)$$

Hat jede Ecke von G den gleichen Grad g so heißt G regulär *oder* g-regulär*.*

Definition 6.1.4 (Teilgraph)
Ein Graph G' ist ***Teilgraph*** *von Graph G (und G ein* Obergraph *von G'), geschrieben $G' \subseteq G$, wenn gilt $E' \subseteq E$ und $K' \subseteq K$.*

Ein gesättigter *Teilgraph G' entsteht aus G durch Entfernen bestimmter Knoten und den inzidenten Kanten. Ein* spannender *Teilgraph G' entsteht aus G durch Entfernen bestimmter Kanten, so dass gilt: $E' = E$ und $K' \subset K$*

Der Teilgraph G' heißt **induziert** *oder* **aufgespannt** *von $E' \in G$, wenn er* **alle** *Kanten $xy \in K(G)$ mit $x, y \in E'(G)$ enthält. Einen solchen induzierten Teilgraphen bezeichnet man als* Untergraphen*. Ein Untergraph hat immer weniger Knoten und Kanten als sein Obergraph und ist nicht gesättigt.*

Definition 6.1.5 (Ordnung)
Die Anzahl der Ecken $|E(G)| = |E|$ in G nennt man die ***Ordnung*** *von G und schreibt dafür auch $|G|$.*

Definition 6.1.6 (Cliquenzahl)
Die größte Mächtigkeit einer Menge von paarweise benachbarten Ecken in G ist die Cliquenzahl $\omega(G)$ *von G.*

Definition 6.1.7 (Isomorphe Graphen)
Zwei Graphen können die gleiche Struktur besitzen, auch wenn sie unterschiedlich definiert sind. Existiert zwischen zwei Graphen G_1 und G_2 eine bijektive Abbildung $f : E(G_1) \to E(G_2)$ zwischen den Eckenmengen beider Graphen, welche die Kanten von G_1 auf diejenigen von G_2 abbildet, so dass gilt

$$(x, y) \in K(G_1) \Leftrightarrow (f(x), f(y)) \in K(G_2),$$

dann nennen wir G_1 und G_2 isomorph *(von gleicher Struktur).*

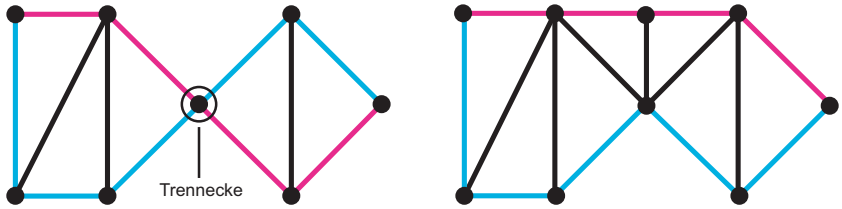


Abbildung 6.3: Einfach und zweifach zusammenhängende Graphen

Definition 6.1.8 (Kantenzüge, Wege und Kreise)
Unter einem ***Kantenzug*** *(oder auch* Pfad*) mit* Anfangspunkt x_1 *und dem* Endpunkt x_n *versteht man eine endliche Folge von Ecken e_i und Kanten k_i der Form $(x_1, k_1, x_2, k_2, \ldots, k_{n-1}, x_n) := P_G(x_1, x_n) = P(x_1, x_n)$, wobei gilt $i \in [1, n-1]$ das k_i mit x_i und x_{i+1} inzidiert. Die Ecken und Kanten dürfen sich in einem Kantenzug wiederholen. Existiert ein Pfad zwischen x und y, dann schreiben wir $P(x, y) \neq \emptyset$ und $P(x, y) = \emptyset$, wenn kein solcher Pfad in G existiert. Ein Kantenzug heißt* ***geschlossen****, wenn $x_1 = x_n$ gilt. Gilt $x_1 \neq x_n$, dann heißt der Kantenzug* ***offen****. Tritt jede Kante in einem Kantenzug genau einmal auf, dann nennt man diesen Kantenzug* ***einfach****. Die Zahl $n := |P(x_1, x_n)|$ gibt die* ***Länge*** *des Kantenzuges an.*

Von einem ***Weg*** *zwischen x_1 und x_n spricht man, wenn die Ecken eines Kantenzuges paarweise verschieden sind.*

Ein geschlossener Kantenzug heißt ***Kreis*** *(oder* ***Zyklus****), wenn $x_1, \ldots, x_{n-1}$ paarweise verschieden sind. Ein Graph heißt* ***kreislos*** *oder* ***azyklisch****, wenn er keinen Kreis enthält.*

Mit Hilfe der Pfadlänge kann eine Distanz zwischen zwei Ecken x und y eines Graphen G, wie folgt definiert werden:

$$\delta_G(x, y) := \begin{cases} \min(\{|P(x, y)| \mid P(x, y) \in G\}), & P(x, y) \neq \emptyset \\ \infty, & P(x, y) = \emptyset \end{cases}$$

In ungerichteten Graphen ist die Distanz kommutativ*, da die Adjazenzbeziehung zwischen den Ecken symmetrisch ist. Dies gilt nicht im allgemeinen für gerichtete Graphen.*

Definition 6.1.9 (Zusammenhang)
Ein nicht leerer Graph heißt zusammenhängend*, wenn er für je zwei seiner Ecken x, y einen $x-y$-Weg enthält. Jeder nicht zusammenhängende Graph lässt sich in maximal zusammenhängende Teilgraphen zerlegen. Diese Teilgraphen nennt man* **Komponenten** *von G. Komponenten können nicht leer sein, deshalb besitzt der leere Graph, als einziger Graph, keine Komponenten. Zerfällt ein Graph G in mindestens zwei Komponenten nach Entfernen einer Ecke, so nennt man diese Ecke eine* **Trennecke** *oder* ***Artikulation*** *von G und bezeichnet diesen Graphen als* einfach zusammenhängend*. Im allgemeinen nennt man einen Graphen G n-fach* zusammenhängend*, wenn je zwei Ecken durch mindestens n kreuzungsfreie Punkte miteinander verbunden sind. Entfernt man aus einem n-fach-zusammenhängenden Graphen höchstens $n-1$ Ecken, so ist der Restgraph noch mindestens einfach-zusammenhängend (siehe Abb.6.3).*

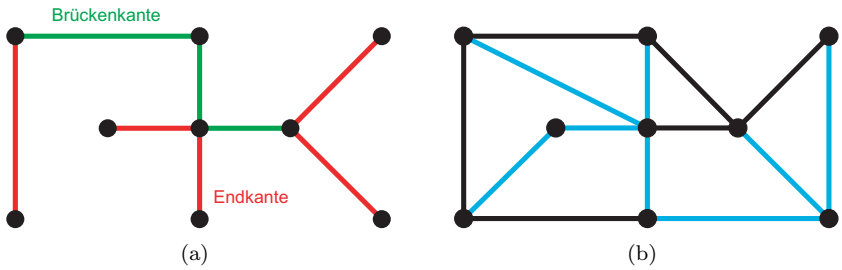


Abbildung 6.4: Beispiel für einen Baum und ein Gerüst

Definition 6.1.10 (Baum und Gerüst)
Unter einem Baum *(Abb.6.4) versteht man einen zusammenhängenden Graphen ohne Kreise. Ein Baum B zeichnet sich durch folgende Eigenschaften aus:*

1. $|K(B)| = |E(B)| - 1$
2. *B ist einfach zusammenhängend.*
3. *Verbindet man zwei nicht adjazente Ecken von B, so entsteht ein Graph B' der genau einen Kreis enthält. Eine solche Verbindung nennt man auch* Sehne.

Ist ein zusammenhängender Graph G kein Baum, so kann man durch Entfernen geeigneter Kanten einen Baum mit der selben Eckenmenge erzeugen. Einen solchen Baum nennt man ein Gerüst *von G. Es können im allgemeinen mehrere Gerüste konstruiert werden, die nicht notwendig isomorph sind, jedoch die gleiche Anzahl von Kanten besitzen.*

Einen azyklischen und nicht zusammenhängenden Graphen nennt man auch ***Wald****.*

6.1.2 Spezielle Typen von Graphen

Endliche Graphen: Ein Graph mit endlicher Eckenmenge heißt *endlich*. Ein endlicher Graph kann eine unendliche Kantenmenge besitzen.

Unendliche Graphen: Ein Graph mit unendlicher Eckenmenge heißt *unendlich*. Ein unendlicher Graph kann eine endliche Kantenmenge besitzen.

Graphen ohne Schlingen und Zweiecke: Verbindet jede Kante zwei unterschiedliche Ecken heißt der Graph *schlingenlos*. Werden je zwei Ecken durch höchstens eine Kante verbunden heißt der Graph *zweiecklos*, siehe Abbildung 6.5a. Es ist natürlich genauso möglich, sogenannte *Multigraphen* zu definieren, die bis zu r Kanten zwischen zwei Ecken besitzen können.

Vollständige Graphen: Besitzt ein Graph keine Schleifen und sind alle Ecken paarweise durch genau eine Kante miteinander verbunden, nennt man den Graphen *vollständig*. Einen vollständigen Graphen auf n Ecken bezeichnen wir mit K^n, siehe Abbildung 6.5b.

Planare (plättbare) Graphen: Ein in der Ebene ohne Überschneidungen von Kanten darstellbarer Graph heißt *plättbar* oder *eben* (Abb. 6.6a, b).

Bipartite Graphen: Lässt sich die Eckenmenge eines Graphen in zwei disjunkte Teilmengen aufteilen, so dass jede Ecke der einen Menge mit mindestens einer Ecke der anderen Menge verbunden ist und gleichzeitig keine Kante existiert, die zwei Ecken der gleichen Teilmenge miteinander verbindet, dann heißt der Graph *bipartit* (Abb. 6.6d).

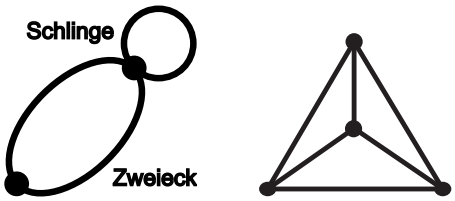
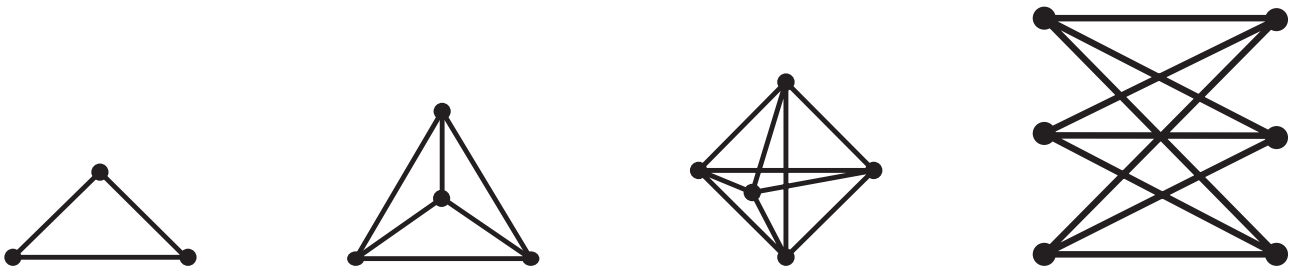


Abbildung 6.5: a) Beispiel für ein Zweieck und eine Schlinge. b) Der vollständige Graph K^4.



(a) K_3 (b) K_4 (c) K_5 (d) $K_{3,3}$

Abbildung 6.6: Abbildungen a) und b) sind vollständige und planare Graphen. Abbildung c) ist ein vollständiger und nicht planarer Graph. Abbildung d) ist ein bipartiter und nicht planarer Graph.

Bewertete (gewichtete) Graphen: Ordnet man jeder Kante einen Wert zu, so spricht man von einem *kanten-bewerteten* oder *kanten-gewichteten* Graphen (Abb. 6.2c). Wird jeder Ecke ein Wert zugeordnet, so spricht man von einem *ecken-bewerteten* oder *ecken-gewichteten* Graphen. Ein *vollständig-bewerteter* Graph ist kanten- und ecken-bewertet. Wenn man von einem bewerteten Graphen spricht, versteht man darunter in den meisten Fällen einen kanten-bewerteten Graphen.

Klassen von Bäumen:

Ketten: Der Grad aller Ecken aus B ist höchstens 2 (Abb. 6.7a).

- *Sterne*: Nur eine Ecke aus B besitzt einen Grad größer als 1 (Abb. 6.7b).
- *Sternketten*: B besitzt eine Kette C, wobei jede Kante von B mit mindestens einer Ecke von C inzidiert (Abb. 6.7c).
- *Spinnen*: Es existiert nur eine Ecke in B mit einem Grad größer als 2.
- *l-Spinnen*: Eine $l-$Spinne (Abb. 6.7d) ist eine Spinne mit *Beinen* der Länge l für die gilt:

$$grad(x) > 2 \text{ und } grad(y) = 1 \Rightarrow |P(x,y)| = l$$

  Sterne sind somit also 1-Spinnen.
- *d-Bäume*: Bäume für die gilt:

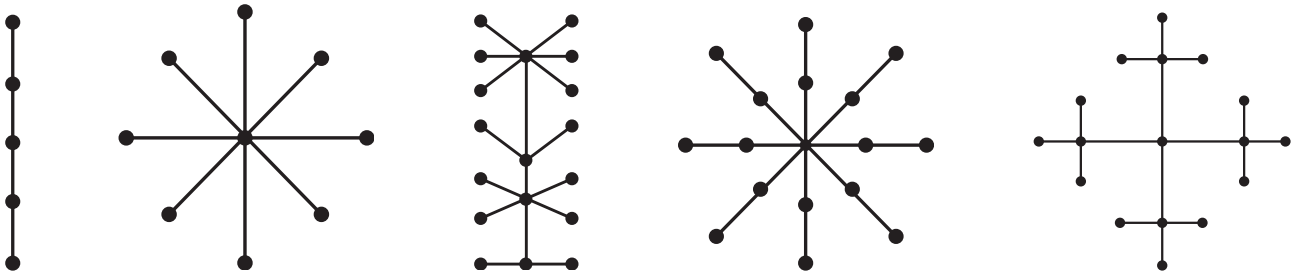
$$\underset{x \in K(G)}{\forall} (grad(x) = d \ \vee \ grad(x) = 1)$$

  nennt man $d-$*Bäume* (Abb. 6.7e).

Es sei angemerkt, dass wir im weiteren dieser Arbeit unter einem Graphen G einen endlichen, schlingen- und zweiecklosen, sowie kanten- oder vollständig-bewerteten Graphen verstehen.

6.1.3 Datenstrukturen für Graphen

Im folgenden wollen wir kurz auf die wesentlichen Datenstrukturen für endliche Graphen eingehen und ihre Vor- und Nachteile aufzählen und folgen dabei (Ottmann & Widmayer 1993). Die Wahl der Datenstruktur hängt



(a) Kette (b) Stern (c) Sternkette (d) 2-Spinne (e) 4-Baum

Abbildung 6.7: Verschiedene Klassen von Bäumen

einerseits von der Effizienz der durchzuführenden Operationen und andererseits vom benötigten Speicherplatz ab. Zeit und Speicherplatz lassen sich in den meisten Fällen jedoch nicht gleichzeitig optimieren, da eine effiziente Laufzeit zu Lasten des Speicherplatzes geht. Ebenfalls ist zu berücksichtigen, ob es sich um eine dynamische oder statische Anwendung handelt. Im Falle einer dynamischen Anwendung wird der Graph durch den Algorithmus verändert und das effiziente Einfügen und Entfernen von Ecken und Kanten stellt ein wichtiges Kriterium dar.

Inzidenzmatrix

Die Inzidenzmatrix sei hier nur zur Vollständigkeit erwähnt. Sie wird jedoch in der Praxis aufgrund ihres hohen Speicherplatzbedarfs nicht verwendet. Die Inzidenzmatrix speichert direkt die Inzidenzbeziehung zwischen den Ecken und den Kanten eines Graphen und da ein vollständiger Graph mit n Ecken ohne Schleifen $n(n-1)$ und mit Schleifen n^2 Kanten besitzt, ist der Speicherbedarf einer Inzidenzmatrix $\Theta(|G|^3)$. Im Falle eines unbewerteten Graphen G erhalten wir so eine *Boole'sche* Matrix $I_G = (i_{ek})$, mit $e \in [1, |G|]$ (Eckenindex) und $k \in [1, |G|^2]$ (Kantenindex), wobei gilt

$$i_{ek} = \begin{cases} 1, & \text{falls } e \text{ mit } k \text{ inzidiert} \\ 0, & \text{sonst} \end{cases}$$

In der Praxis wird anstatt der Inzidenzmatrix die sogenannte Adjazenzmatrix verwendet. Jedoch lassen sich Inzidenz- und Adjazenzmatrix durch einfache Matrizenmultiplikation ineinander überführen (Bill & Fritsch 1991).

Adjazenzmatrix

Ein Graph $G = (E, K)$ kann in einer $|E| \times |E|$ Matrix $A_G = (a_{ij})$ gespeichert werden, welche die Adjazenzbeziehungen der Ecken von G enthält und somit auch implizit die Inzidenzbeziehungen der Ecken und Kanten und benötigt dabei weniger Speicher als die Inzidenzmatrix. Die Indizes i und j stehen dabei jeweils für eine Ecke aus G, indem wir die Ecken von G von 1 bis $|E|$ (oder 0 bis $|E| - 1$) durchnummerieren. Im Falle eines ungerichteten Graphen ist A_G symmetrisch. Handelt es sich um einen unbewerteten Graphen G, so ist A_G eine *Boole'sche* Matrix, d.h.

$$a_{ij} = \begin{cases} 1, & \text{falls } ij \in K(G) \\ 0, & \text{sonst} \end{cases}$$

Ist der Graph G dagegen bewertet, dann ist

$$a_{ij} = \begin{cases} w_{ij}, & \text{falls } ij \in K(G) \\ 0, & \text{sonst} \end{cases}$$

Da die Werte eines bewerteten Graphen nicht Zahlen sein müssen, wird der Wert 0 in diesem Zusammenhang dazu verwendet, um eine nicht vorhandene Kante darzustellen.

Eine Adjazenzmatrix A_G hat die folgenden Eigenschaften:

- Der Speicheraufwand ist $\Theta(|G|^2)$.
- Der Speicheraufwand ist unabhängig von der Anzahl der Kanten in G. Daraus folgt sofort, dass Adjazenzmatrizen bzgl. Speicher ungünstig sind, wenn der Graph vergleichsweise wenige Kanten enthält.
- Der Test auf Adjazenz zweier Ecken benötigt $O(1)$ Zeit und ist optimal. Viele Algorithmen erfordern jedoch eine Initialisierung der Matrix oder die Berücksichtigung aller Einträge der Matrix und benötigen deshalb $\Omega(|G|^2)$ Rechenschritte. Durch geeignete Zusatzinformationen lässt sich jedoch in vielen Fällen Abhilfe schaffen, ohne dabei den Speicherbedarf über $\Theta(|G|^2)$ zu erhöhen. Typische Operationen, wie das Inspizieren aller zu einer gegebenen Ecke inzidenten Kanten sind für Graphen mit wenigen Kanten ineffizient, genauso wie das Einfügen einer neuen Ecke in einen Graphen mit vielen Ecken.

Inzidenzliste

Die *Inzidenzliste* oder auch *doppelt verkettete Ecken-Kanten-Liste* ist wohl die am meisten verwendete Datenstruktur für Graphen, da sie $\Theta(|E(G)| + |K(G)|)$ Speicherplatz benötigt und viele Operationen, wie z.B. das Traversieren durch einen Graphen, die Suche in einem Baum und insbesondere das Einfügen ($O(1)$) und Entfernen ($O(n)$) von Knoten und Kanten sehr gut unterstützt.

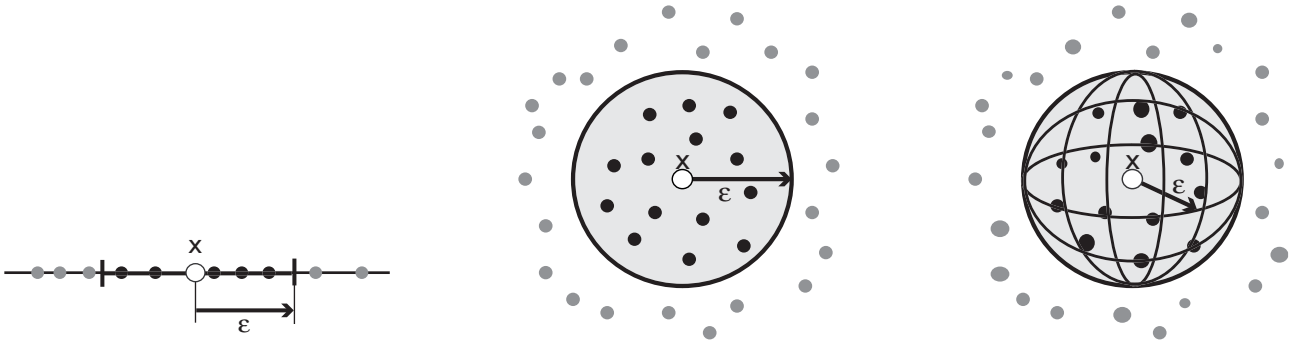
Die Datenstruktur einer Inzidenzliste enthält einerseits eine doppelt verkettete Liste aller Ecken von G und für jede Ecke einen Verweis auf eine doppelt verkettete Liste mit allen inzidenten Kanten. Jede Kante wiederum besitzt noch einen Verweis auf ihre Start- und Endecke. Im Englischen wird diese Datenstruktur auch *doubly connected arc list (DCAL)* genannt.

6.2 Typen von Nachbarschaftsgraphen

Eine allgemeine Einführung in das Gebiet der Nachbarschaftsgraphen findet man in (Jaromczyk & Toussaint 1992). Nachbarschaftsgraphen werden auch als *Proximity Graphs* (Toussaint 1991) bezeichnet. Sie finden überall dort Anwendung, wo Form und Struktur von Punktmengen von Interesse sind, wie z.B. Computer Vision, Mustererkennung (Zahn 1971), Algorithmische Morphologie (Kirkpatrick & Radke 1985), Kartographie, Geographie und Biologie. In Nachbarschaftsgraphen werden *ähnliche* oder *benachbarte* Punkte mit einer Kante verbunden. Die vielfältigen Möglichkeiten der Definition von *benachbart* führen zu mehreren verwandten Graphen von denen die bekanntesten hier aufgezählt sind:

- Nächster Nachbar Graph,
- K-Nächster Nachbar Graph,
- Minimaler spannender Baum,
- Geographischer Nachbarschaftsgraph,
- Relativer Nachbarschaftsgraph,
- Gabriel Graph,
- β-Skelette,
- Delaunay-Triangulation,
- Urquhart Graph,
- Einflussbereichsgraph,

Bevor wir auf die einzelnen Nachbarschaftsgraphen näher eingehen, führen wir noch die folgenden Schreibweisen und Bezeichnungen ein:



(a) Intervall (b) Kreis (c) Kugel

Abbildung 6.8: Veranschaulichung der ϵ-Umgebung im $\mathbb{R}^1$ a), $\mathbb{R}^2$ b) und $\mathbb{R}^3$ c).

Definition 6.2.1 (ϵ-Umgebung)
Sei **P** *eine Menge von n Punkten im* $\mathbb{R}^d$ *und* $\delta(p,q)$ *eine beliebige Metrik auf* $\mathbb{R}^d$*, dann bezeichnet*

$$\mathcal{U}_{P,\delta}(p,\epsilon) = \{q \in P \mid \delta(p,q) < \epsilon\}$$

die ϵ*-Umgebung eines Punktes in* **P**. *Im folgenden kurz mit* $\mathcal{U}(p,\epsilon)$ *bezeichnet.*

Die ϵ-Umgebung lässt sich anhand des $\mathbb{R}^n$ einfach veranschaulichen (siehe Abb. 6.8). Im $\mathbb{R}^1$ ist die ϵ-Umgebung ein offenes Intervall, im $\mathbb{R}^2$ die offene Kreisfläche und im $\mathbb{R}^3$ das offene Kugelvolumen. Im $\mathbb{R}^n$ kann die ϵ-Umgebung als n-dimensionale *Hypersphäre* betrachtet werden.

Definition 6.2.2 (Nächster Nachbar)
Sei **P** *eine Menge von n Punkten im* $\mathbb{R}^d$ *und* $\delta(p,q)$ *eine beliebige Metrik auf* $\mathbb{R}^d$*, dann heißt* $q = \mathcal{NN}_{\mathbf{P}}(p)$ nächster Nachbar *von p, wenn gilt:*

$$\mathcal{U}(p,\delta(p,q)) \cap P = \emptyset.$$

Definition 6.2.3 (Linse zweier Punkte)
Sei **P** *eine Menge von n Punkten im* $\mathbb{R}^d$ *und* $\delta(p,q)$ *eine beliebige Metrik auf* $\mathbb{R}^d$*, dann bezeichnet*

$$\mathcal{L}(p,q) = \mathcal{U}(p,\delta(p,q)) \cap \mathcal{U}(q,\delta(p,q))$$

das linsenförmige *Gebiet zwischen den Punkten p und q. Im Englischen wird diese Menge mit* **Lune** *bezeichnet.*

Definition 6.2.4 ($\beta-$Linse zweier Punkte)
Sei **P** *eine Menge von n Punkten im* $\mathbb{R}^d$ *und* $\delta(p,q)$ *eine beliebige Metrik auf* $\mathbb{R}^d$*, dann bezeichnet*

$$\mathcal{L}_\beta(p,q) = \mathcal{U}(\frac{2p+\beta(q-p)}{2}, \frac{\beta\delta(p,q)}{2}) \cap \mathcal{U}(\frac{2q+\beta(p-q)}{2}, \frac{\beta\delta(p,q)}{2})$$

die verallgemeinerte $\beta-$*Linse zwischen den Punkten p und q (Abb. 6.9).*

6.2.1 Nächster Nachbar Graph

Der wohl bekannteste Nachbarschaftsgraph mit vielen Anwendungen auf Gebieten wie dem Data Mining (Clustering, Klassifizierung), dem maschinellen Lernen oder der Mustererkennung ist der Nächste-Nachbar-Graph (englisch: *All Nearest Neighbour Graph*).

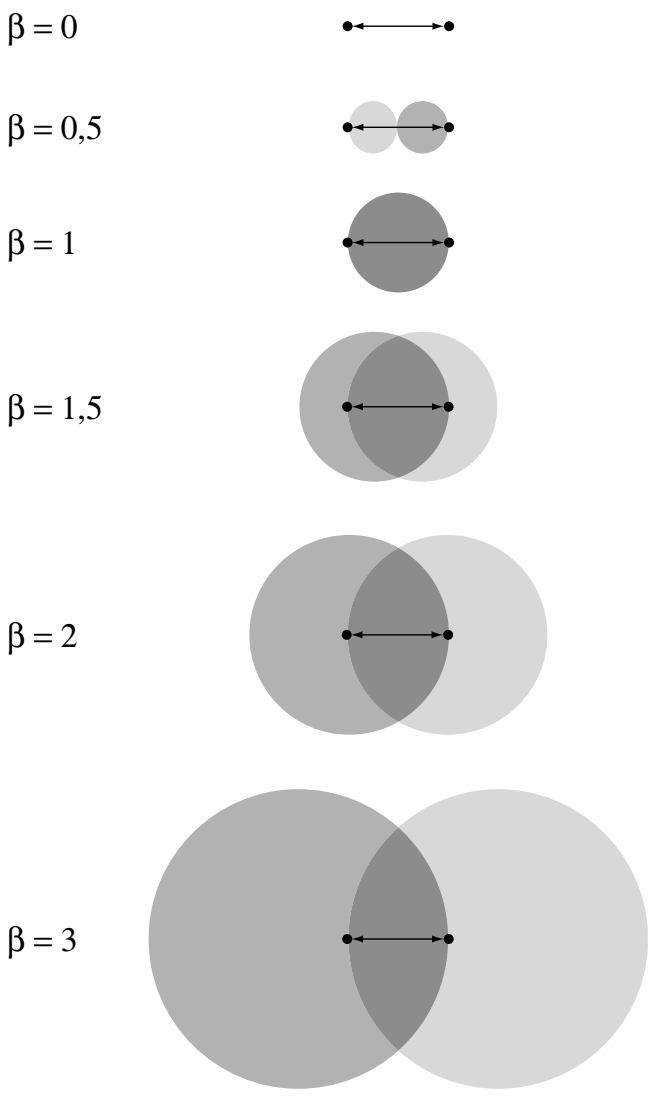


Abbildung 6.9: Beta Lunes

Definition 6.2.5 (Nächster-Nachbar-Graph – NNG)
Sei $\mathbf{P}$ *eine Menge von* n *Punkten im* $\mathbb{R}^d$ *und* $\delta(p,q)$ *eine beliebige Metrik auf* $\mathbb{R}^d$*, dann bezeichnet man mit* $\mathsf{NNG}(\mathbf{P})$ *den* Nächsten- Nachbar-Graphen *der Menge* $\mathbf{P}$*, wenn folgende Bedingung erfüllt ist:*

$$\mathsf{NNG}(P) = G(P,K) \text{ mit } K = \{pq \mid p,q \in P \ \wedge \ (\ \mathcal{U}(p,\delta(p,q)) \ \cap \ P = \emptyset \ \vee \ \mathcal{U}(q,\delta(p,q)) \ \cap \ P = \emptyset \ )\} \tag{6.1}$$

Der Nächste-Nachbar-Graph (Abb. 6.11 (b) und C.2 (b))[5] lässt sich verallgemeinern, indem nicht nur die Nachbarn mit der kleinsten Distanz zulässig sind, sondern auch Nachbarn mit den nächsten $k-1$ größeren Distanzen. Den auf diese Weise erhaltenen Graphen nennt man den sogenannten ***k-Nächster-Nachbar-Graph*** (Abbildung 6.12 (a), (b) und (c)), den man formal wie folgt definieren kann.

Definition 6.2.6 (k-Nächster-Nachbar-Graph – k-NNG)
Sei $\mathbf{P}$ *eine Menge von* n *Punkten im* $\mathbb{R}^d$ *und* $\delta(p,q)$ *eine beliebige Metrik auf* $\mathbb{R}^d$*, dann bezeichnet man mit* $\mathsf{k-NNG}(\mathbf{P})$ *den* k-Nächster-Nachbar-Graphen *der Menge* $\mathbf{P}$*, wenn folgende Bedingung erfüllt ist:*

$$\mathsf{k-NNG}(P) = G(P,K) \text{ mit } K = \{pq \mid p,q \in P \ \wedge \ (\ |\mathcal{U}(p,\delta(p,q)) \cap P| < k \ \vee \ |\mathcal{U}(q,\delta(p,q)) \cap P| < k \ )\} \tag{6.2}$$

Von der zahlreichen Literatur zu Nächsten Nachbargraphen und ihren Anwendungen seien hier genannt (Eppstein, Paterson & Yao 1997, Nakano & Olariu 1997, Jarvis & Patrick 1973).

6.2.2 Minimaler spannender Baum

In Netzwerken ist ein häufiges Problem, n Orte (Ecken) so miteinander zu verbinden, dass die *Kosten* (abhängig von der Art des Problems) für das Netzwerk minimal werden. Wie der Name es schon ausdrückt, handelt es sich also bei diesem Graphen um einen Baum und somit um einen zusammenhängenden und azyklischen Graphen.

Definition 6.2.7 (Minimaler spannender Baum – MST (Minimum Spanning Tree))
Sei $\mathbf{P}$ *eine Menge von* n *Punkten im* $\mathbb{R}^d$ *und* $\delta(p,q)$ *eine beliebige Metrik auf* $\mathbb{R}^d$*, dann bezeichnet man mit* $\mathsf{MST}(\mathbf{P})$ *einen* minimal spannenden Baum *der Menge* $\mathbf{P}$*, wenn folgende Bedingung erfüllt ist:*

$$\mathsf{MST}(P) = G(P,K) \text{ ist ein Gerüst von } P \text{ und } \sum_{pq \in K} \delta(p,q) \stackrel{!}{=} \min \tag{6.3}$$

Effiziente Methoden zu Berechnung von minimal spannenden Bäumen werden zum Beispiel in (Yao 1982, Supowit 1983, King 1995) beschrieben. Die Abbildungen 6.11 (c) und C.3 (c) zeigen zwei Beispiele für einen minimal spannenden Baum.

6.2.3 Relativer Nachbarschaftsgraph

Toussaint (1980*b*) führte 1980 den *Relative Neighbourhood Graph*, den wir mit Relativer Nachbarschaftsgraph übersetzen, ein, um Probleme bei der Strukturanalyse von Punktmengen zu lösen.

[5] Beispiele für den NNG sowie für den kNNG, den MST, den RNG, den GG, die DT und den SIG finden sich in den Abbildungen 6.11 und 6.12 und im Anhang C.

Definition 6.2.8 (Relativer Nachbarschaftsgraph – RNG)
Sei $\mathbf{P}$ *eine Menge von* n *Punkten im* $\mathbb{R}^d$ *und* $\delta(p,q)$ *eine beliebige Metrik auf* $\mathbb{R}^d$*, dann bezeichnet man mit* $\mathsf{RNG}(\mathbf{P})$ *den* Relativer Nachbarschaftsgraph *der Menge* $\mathbf{P}$*, wenn folgende Bedingung erfüllt ist:*

$$\mathsf{RNG}(P) = G(P,K) \text{ mit } K = \{pq \mid p,q \in P \ \wedge \ \mathcal{L}_2(p,q) \ \cap \ P = \emptyset\} \tag{6.4}$$

Eine äquivalente Definition für die Kantenmenge lautet:

$$K = \{pq \mid p,q \in P \ \wedge \ \underset{r \in P \setminus \{p,q\}}{\forall} (\delta(p,q) \ \leq \ \max\{\delta(p,r), \delta(q,r)\})\} \tag{6.5}$$

Der $\mathsf{RNG}(P)$ (siehe Abb. 6.11 (d) und C.3 (d)) ist ein Obergraph des $\mathsf{MST}(P)$ und deshalb zusammenhängend.

Der $\mathsf{RNG}(P)$ lässt sich in gleicher Weise wie der $\mathsf{NNG}(P)$ zum $\mathsf{k-RNG}(P)$ verallgemeinern. Weitere Informationen zum relativen Nachbarschaftsgraph findet man auch in (Toussaint 1980*a*, Supowit 1983, Agarwal & Matousek 1992, Jaromczyk & Toussaint 1992, Lingas 1994).

6.2.4 Geographischer Nachbarschaftsgraph

Der *Geographische Nachbarschaftsgraph* $\mathsf{GNG}(\mathbf{P})$ (Yao 1982, Rao 1998, Nakano & Olariu 1997) wird durch sogenannte *Narrow Regions* $\mathcal{R}$ definiert. Wir geben hier die Definition für den 2-dimensionalen Raum an. Die Verallgemeinerung auf den n-dimensionalen Raum findet man z.B. in (Yao 1982) und (Rao 1998).

Die Umgebung eines Punktes $p \in P = \mathbb{R}^2$ kann, wie in Abbildung 6.10 dargestellt, in acht Regionen (Narrow Regions) $\mathcal{R}_{1\ldots8}$ eingeteilt werden, so dass gilt: $P = \bigcup_{1 \leq i \leq 8} \mathcal{R}_i(p)$. Der $\mathsf{GNG}(\mathbf{P})$ definiert sich dann wie folgt:

Definition 6.2.9 (Geographischer Nachbarschaftsgraph – GNG)
Sei $\mathbf{P}$ *eine Menge von* n *Punkten im* $\mathbb{R}^2$ *und* $\delta(p,q)$ *eine beliebige Metrik auf* $\mathbb{R}^2$*, dann bezeichnet man mit* $\mathsf{GNG}(\mathbf{P})$ *den* Geographische Nachbarschaftsgraph *der Menge* $\mathbf{P}$*, wenn folgende Bedingung erfüllt ist:*

$$\mathsf{GNG}(P) = G(P,K) \text{ mit } K = \bigcup_{1 \leq i \leq 8} \{pq \mid p \in P \ \wedge \ q \in \mathcal{NN}_{\mathcal{R}_i(p)}(p)\}$$

D.h. in den 8 Regionen werden jeweils die nächsten Nachbarn identifiziert, die dann schließlich den $\mathsf{GNG}(P)$ bilden. Der $\mathsf{GNG}(P)$ ist ein Obergraph des $\mathsf{RNG}(P)$ und kann auch wie dieser zum $\mathsf{k-GNG}(P)$ verallgemeinert werden.

6.2.5 Gabriel Graph

Der Gabriel Graph wurde durch Gabriel und Sokal (Gabriel & Sokal 1969) zur geographischen Variationsanalyse eingeführt.

Definition 6.2.10 (Gabriel Graph – GG)
Sei $\mathbf{P}$ *eine Menge von* n *Punkten im* $\mathbb{R}^d$ *und* $\delta(p,q)$ *eine beliebige Metrik auf* $\mathbb{R}^d$*, dann bezeichnet man mit* $\mathsf{GG}(\mathbf{P})$ *den* Gabriel Graph *der Menge* $\mathbf{P}$*, wenn folgende Bedingung erfüllt ist:*

$$\mathsf{GG}(P) = G(P,K) \text{ mit } K = \{pq \mid p,q \in P \ \wedge \ \mathcal{L}_1(p,q) \ \cap \ P = \emptyset\} \tag{6.6}$$

Eine äquivalente Definition für die Kantenmenge lautet:

$$K = \{pq \mid p,q \in P \ \wedge \ \underset{r \in P \setminus \{p,q\}}{\forall} (\delta(p,q)^2 \ \leq \ \delta(p,r)^2 + \delta(q,r)^2)\} \tag{6.7}$$

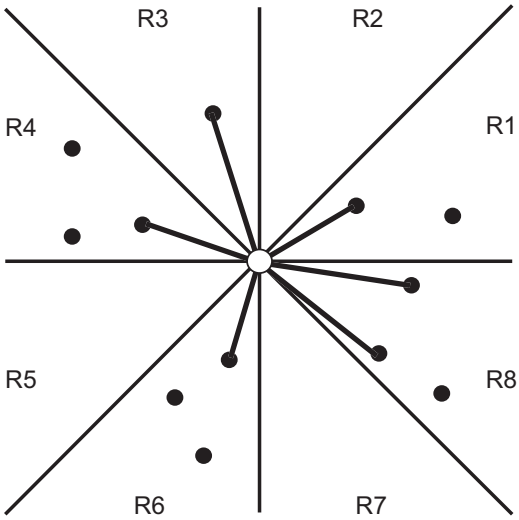


Abbildung 6.10: *Narrow Regions* eines Punktes p (Zentrum in der Abb.) im $\mathbb{R}^2$ mit seinen jeweiligen nächsten Nachbarn.

Der $\mathsf{GG}(P)$ ist wie der $\mathsf{RNG}(P)$ zusammenhängend und kann auch wie dieser zum $\mathsf{k}-\mathsf{GG}(P)$ verallgemeinert werden. Zwei Beispiele für den Gabriel Graphen zeigen die Abbildungen 6.11 (e) und C.4 (a).

6.2.6 $\beta-$Skelette

Kirkpatrick & Radke (1985) führte eine Klasse von Graphen ein, indem er das Prinzip der $\beta-Linse$ (Def. 6.2.4) einführte, die eine Verallgemeinerung der Bedingung des RNG darstellt. $\beta-\mathsf{Skelette}$ eignen sich zur Analyse der inneren Struktur von Punktmengen.

Definition 6.2.11 ($\beta-$Skelette – S_β)
Sei $\mathbf{P}$ *eine Menge von* n *Punkten im* $\mathbb{R}^d$ *und* $\delta(p,q)$ *eine beliebige Metrik auf* $\mathbb{R}^d$*, dann bezeichnet man mit* $\mathsf{S}_\beta(\mathbf{P})$ *das* $\beta-$Skelett *der Menge* $\mathbf{P}$*, wenn folgende Bedingung erfüllt ist:*

$$\mathsf{S}_\beta(P) = G(P,K) \text{ mit } K = \{pq \mid p,q \in P \ \wedge \ \mathcal{L}_\beta(p,q) \ \cap \ P = \emptyset\} \tag{6.8}$$

Eine ausführliche Beschreibung dieser Graphen und ihrer effizienten Berechnung, sowie die Erweiterung auf sogenannte $\mathsf{k}\beta-\mathsf{Skelette}$ findet man in (Rao 1998).

6.2.7 Delaunay-Triangulation

Die *Delaunay-Triangulation* (Preparata & Shamos 1988) wird traditionell als dualer Graph des *Voronoi Diagramms* ((Lee 1980)) definiert. (O'Rourke 1982) zeigte jedoch, dass für die L_1 und die L_∞ Norm der duale Graph des Voronoi Diagramms nicht notwendigerweise ein Obergraph des RNG ist und schlug die folgende Definition vor, die direkt auf der Menge der Punkte aufbaut.

Definition 6.2.12 (Delaunay-Triangulation – DT)

$$\mathsf{DT}(P) = G(P,K) \text{ mit } K = \{pq \mid p,q \in P \ \wedge \underset{x\in\mathbb{R}^d, r\in\mathbb{R}^+}{\exists} (p,q \in \partial\mathcal{U}(x,r) \ \wedge \ \mathcal{U}(x,r) \ \cap \ P = \emptyset)\} \tag{6.9}$$

Diese Definition ist äquivalent zur traditionellen Definition im $\mathbb{R}^2$ für alle L_p Normen, für die gilt $1 < p < \infty$ ((Lee 1980, O'Rourke 1982)) und erlaubt die Erweiterung der Teilmengenbeziehung $\mathsf{RNG}(P) \subseteq \mathsf{DT}(P)$ auf die

Normen L_1 und $L\infty$ für beliebige Dimensionen. Die Abbildungen 6.11 (f) und C.4 (b) zeigen zwei Beispiele für die Delaunay-Triangulation.

6.2.8 Urquhart Graph

Entfernt man aus jedem Dreieck einer Delaunay-Triangulation einer Punktmenge $\mathbf{P}$ die jeweils längste Kante, so erhält man den sogenannten *Urquhart Graph* $\mathsf{UG}(P)$ (Urquhart 1980). Urquhart schlug diese Methode vor, um den $\mathsf{RNG}(P)$ effizient zu berechnen, jedoch fand Toussaint (Toussaint 1980*a*) ein Gegenbeispiel, das zeigte, dass der $\mathsf{UG}(P)$ im allgemeinen ein Obergraph des RNG ist. Der $\mathsf{UG}(P)$ ist jedoch eine sehr gute Approximation des RNG, da er nur wenige zusätzliche Kanten enthält.

Formal lässt sich der Urquhart Graph unter Einbeziehung der Delaunay-Triangulation, wie folgt definieren:

Definition 6.2.13 (Urquhart Graph – UG)

$$\mathsf{UG}(P) = G(P,K) \text{ mit } K = \{pq \mid p,q \in P \wedge \underset{r \in P \setminus \{p,q\}}{\forall} (pr, qr \in \mathsf{DT}(P) \wedge \delta(p,q) < \max(\delta(p,r), \delta(q,r)))\} \quad (6.10)$$

Mit anderen Worten, die Bedingung des $\mathsf{RNG}(P)$ wird nicht mehr für alle Punkte $r \in \mathbf{P} \setminus p,q$ gefordert, sondern nur noch für alle r die adjazent zu p und q in $\mathsf{DT}(P)$ sind.

6.2.9 Einflussbereichsgraph

Im Zusammenhang mit der Analyse von Punktmustern schlug Toussaint (Toussaint 1988) als weiteren Graphen den sogenannten *Sphere of Influence Graphen* vor, den wir mit Einflussbereichsgraphen übersetzen.

Definition 6.2.14 (Einflussbereichsgraph – SIG)
Sei $\mathbf{P}$ *eine Menge von* n *Punkten im* $\mathbb{R}^d$ *und* $\delta(p,q)$ *eine beliebige Metrik auf* $\mathbb{R}^d$*, dann bezeichnet man mit* $\mathsf{SIG}(\mathbf{P})$ *den* Einflussbereichsgraph *der Menge* $\mathbf{P}$*, wenn folgende Bedingung erfüllt ist:*

$$\mathsf{SIG}(P) = G(P,K) \text{ mit } K = \{pq \mid p,q \in P \wedge \mathcal{U}(p, \delta(p, \mathcal{NN}_{\mathbf{P}}(p))) \cap \mathcal{U}(q, \delta(q, \mathcal{NN}_{\mathbf{P}}(q))) \neq \emptyset\} \quad (6.11)$$

Der $\mathsf{SIG}(P)$ muss im allgemeinen weder zusammenhängend noch planar sein. Zwei Beispiele für den $\mathsf{SIG}(P)$ zeigen die Abbildungen 6.12 (d) und C.4 (c).

6.3 Hierarchie der Nachbarschaftsgraphen

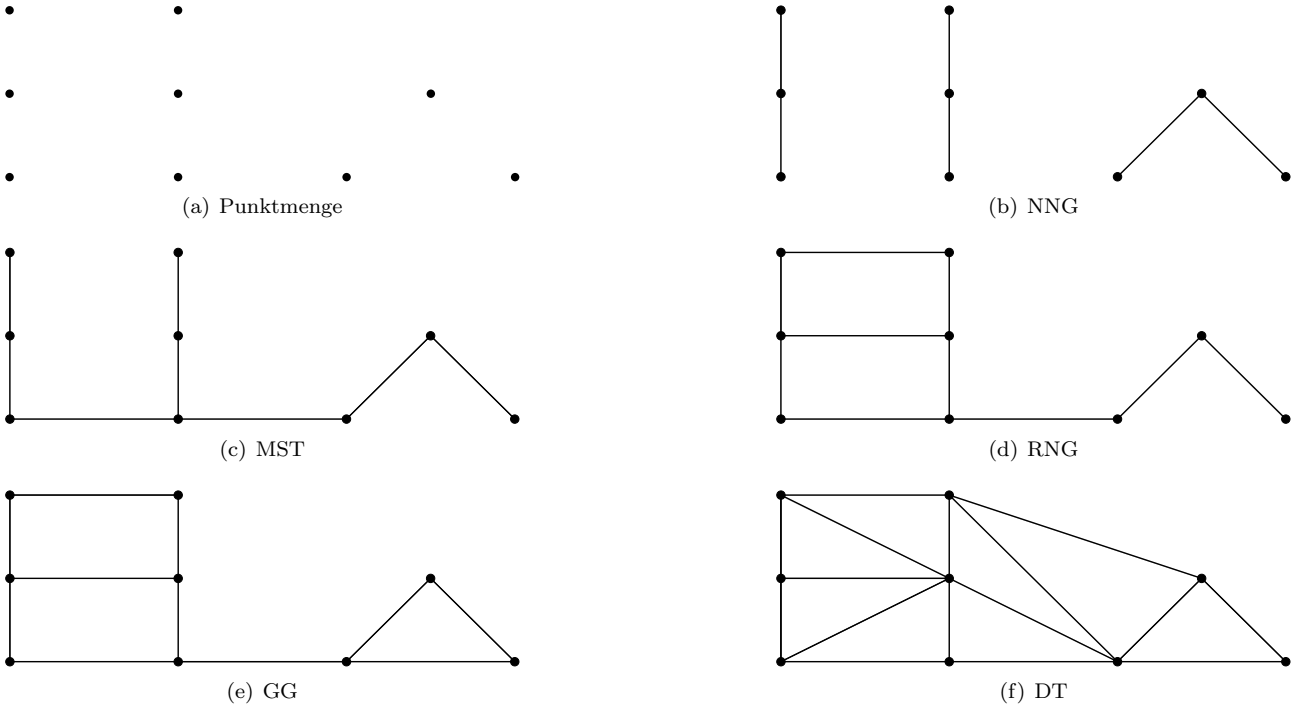
Zwischen den verschiedenen Nachbarschaftsgraphen existieren Teilmengenbeziehungen, so gilt im $\mathbb{R}^n$ für eine beliebige L_p Metrik mit $1 < p < \infty$ die folgende Teilmengenbeziehung (siehe Seite 802 in (Goodman & O'Rourke 1997)).

$$\mathsf{NNG} \subseteq \mathsf{MST} \subseteq \mathsf{RNG} \subseteq \mathsf{GG} \subseteq \mathsf{DT} \quad (6.12)$$

Gleichung 6.12 kann für ein beliebiges $\beta \in [1,2]$ folgendermaßen erweitert werden:

$$\mathsf{NNG} \subseteq \mathsf{MST} \subseteq \mathsf{RNG} \subseteq \mathsf{S}_\beta \subseteq \mathsf{GG} \subseteq \mathsf{DT} \quad (6.13)$$

Für unser, im Kapitel 7, beschriebenes parameterfreies Clusteringverfahren wird Gleichung 6.12 die wesentliche Grundlage bilden, um eine Art parameterfreies *Buffer Growing* zu definieren, dagegen kommt Gleichung 6.13 aufgrund des Parameters β für uns nicht in Frage.

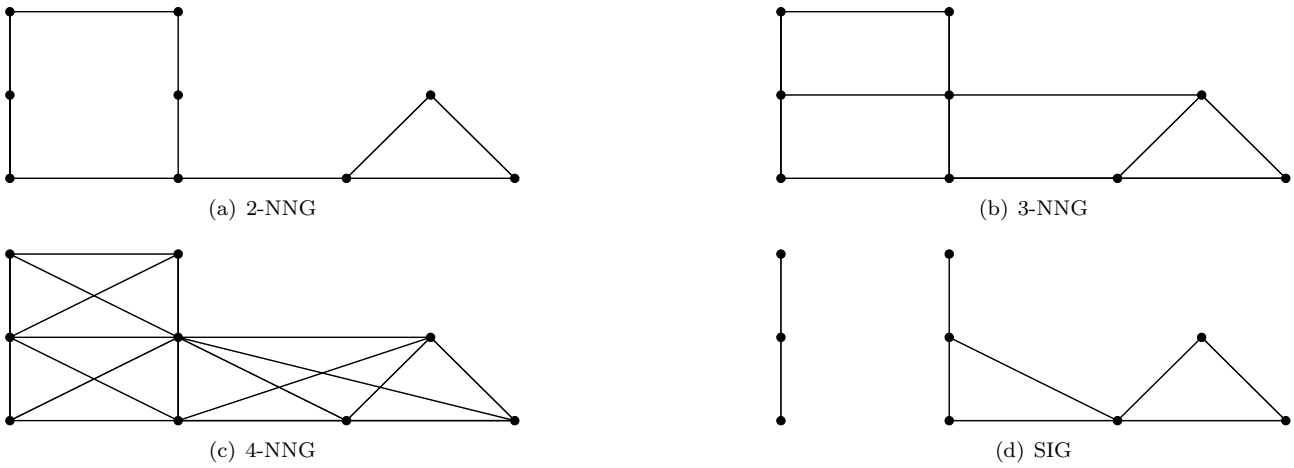


(a) Punktmenge (b) NNG

(c) MST (d) RNG

(e) GG (f) DT

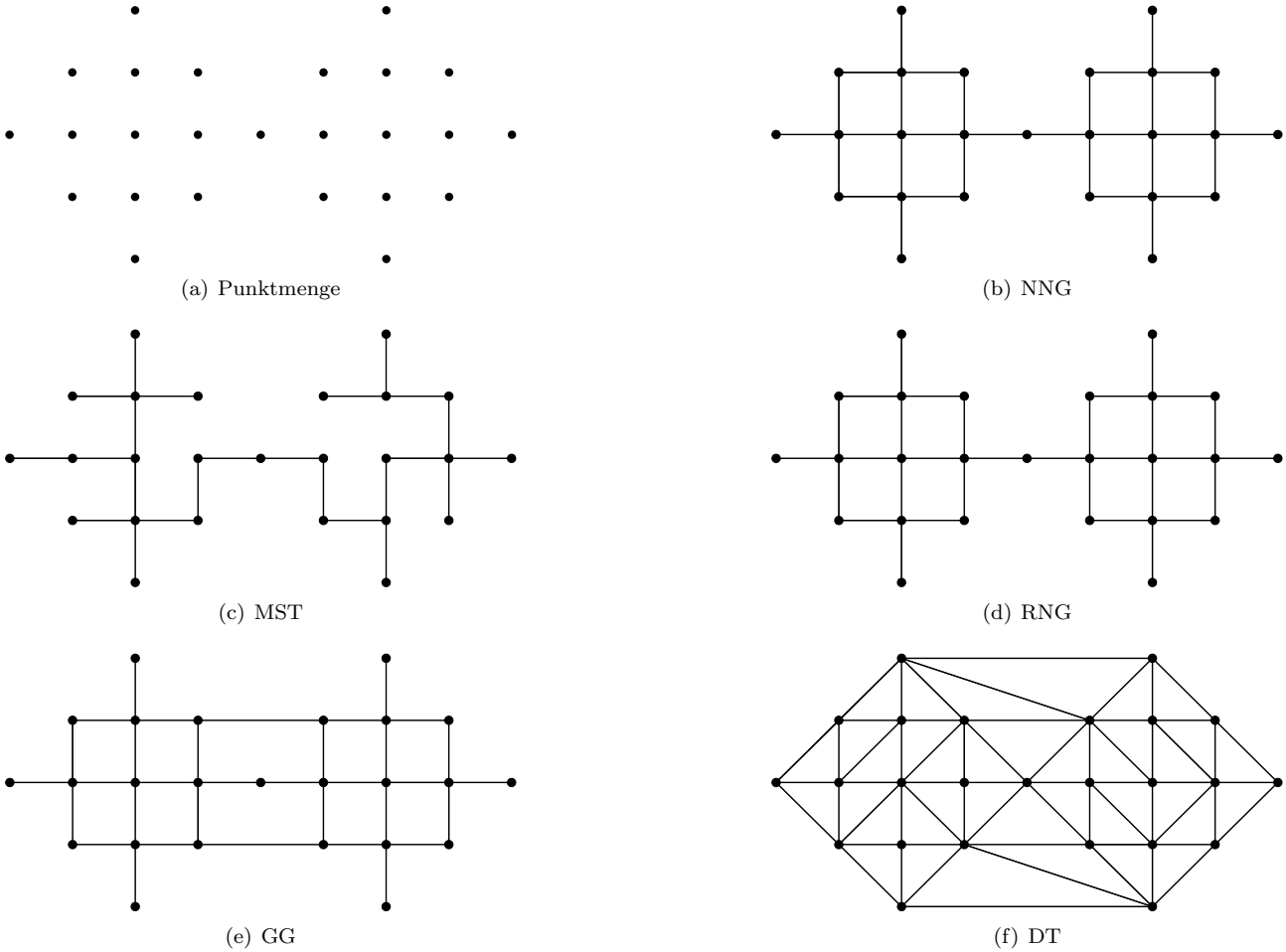
Abbildung 6.11: Visualisierung der Teilmengenbeziehung in der Hierarchie der Nachbarschaftsgraphen



(a) 2-NNG (b) 3-NNG

(c) 4-NNG (d) SIG

Abbildung 6.12: Die k-NNG bilden für sich eine Teilmengen-Hierarchie, aber nicht mit den anderen Graphen aus Abbildung 6.11. Vergleicht man (d) mit 6.11 (f), dann erkennt man, dass in diesem Fall der SIG eine Teilmenge der Delaunay-Triangulation ist, was jedoch nicht im allgemeinen gilt.



(a) Punktmenge (b) NNG

(c) MST (d) RNG

(e) GG (f) DT

Abbildung 6.13: Verletzung der Teilmengenbeziehung im Falle äquidistanter Punkte: $MST \subset NNG$ und $NNG = RNG$. Die inneren diagonalen Kanten der DT (f) sind ebenfalls nicht eindeutig, was im Falle äquidistanter Punkte immer gilt.

Es sei angemerkt, dass die Beziehung $\mathsf{NNG} \subseteq \mathsf{MST}$ in Gleichung 6.12 und 6.13 nur für Punktmengen in allgemeiner Lage gilt. „In allgemeiner Lage“ bedeutet, dass alle Graphen eindeutig sind und zu jedem Punkt p genau ein nächster Nachbar existiert. In Abbildung 6.13 wird verdeutlicht, dass die Teilmengenbeziehung $\mathsf{NNG} \subseteq \mathsf{MST}$ nicht im allgemeinen gilt, wenn mehrere nächste Nachbarn (mehrere Punkte mit dem gleichen Abstand) zu einem Punkt p existieren. Diese Ausnahmen bilden jedoch kein Problem für unser parameterfreies Clusterverfahren, da es höchstens bedeutet, dass sich die Menge der Nachbarschaftsbeziehungen beim Übergang vom NNG zum MST nicht vergrößert und deshalb können wir auf die Bedingung der allgemeinen Lage der betrachteten Punkte verzichten.

Die gültige Teilmengenbeziehung aus Gleichung 6.14 lassen wir in unserem Clusterverfahren ebenfalls außer Betracht, da der *Urquhart Graph* $\mathsf{UG}(P)$, wie oben beschrieben, sich nur geringfügig vom *Relativen Nachbarschaftsgraphen* $\mathsf{RNG}(P)$ unterscheidet und somit keine relevante Zwischenmenge zum *Gabriel Graphen* $\mathsf{GG}(P)$ darstellt.

$$\mathsf{NNG} \subseteq \mathsf{MST} \subseteq \mathsf{RNG} \subseteq \mathsf{UG} \subseteq \mathsf{GG} \subseteq \mathsf{DT} \tag{6.14}$$

6.4 Komplexität

Nachbarschaftsgraphen können höchstens n^2 Kanten besitzen. In diesem Fall würde es sich um einen vollständigen Graphen handeln. $\Theta(n^2)$ ist deshalb die kleinste obere Schranke für den Speicherplatzbedarf für alle Nachbarschaftsgraphen bzgl. jeder beliebigen Metrik und Dimension. Die Delaunay-Triangulation ist der Obergraph der anderen von uns verwendeten Graphen (NNG, MST, RNG, GG) und für jede beliebige feste Dimension d ist $\Theta(n^{\lceil d/2 \rceil})$ die kleinste obere Schranke für den Zeitbedarf zur Berechnung der Delaunay-Triangulation und deshalb kann man auch die Untergraphen mit dieser Schranke nach oben abschätzen. Da immer drei Objekte bei der Berechnung des RNG und des GG miteinander verglichen werden müssen, ergibt sich für diese Graphen sofort ein trivialer Algorithmus mit einem Zeitbedarf von $O(n^3)$. Dies gilt für jede beliebige Metrik und für jede beliebige Dimension. Bei der Berechnung des NNG und des MST müssen dagegen immer nur 2 Objekte miteinander verglichen werden und somit ergibt sich eine obere Schranke von $O(n^2)$. Es existieren jedoch für unterschiedliche Metriken und Dimensionen bessere Schranken, von denen einige in den Tabellen 6.1 und 6.2 aufgelistet sind.

Graph	Dimension	Metrik	Anzahl der Kanten
NNG	≥ 2	L_p	$O(n)$
k-NNG	≥ 2	L_p	$O(n)$
MST	2	$L_p, 1 < p < \infty$	$O(n)$
		L_1, L_∞	$O(n^2)$
	> 2	L_p	$O(n^2)$
RNG	2	$L_p, 1 < p < \infty$	$O(n) :\in [n-1, 3n-6]$
	≥ 2	L_1, L_∞	$O(n^2)$
	3	L_2	$O(n^{4/3})$
	≥ 3	L_p	$O(n^2)$
GG	2	$L_p, 1 < p < \infty$	$O(n)$
	≥ 2	L_1, L_∞	$O(n^2)$
	≥ 3	L_p	$O(n^2)$
DT	2	L_p	$O(n)$
	≥ 3	L_p	$O(n^2)$

Tabelle 6.1: Größenordnung der Kantenmenge von Nachbarschaftsgraphen.

Graph	Dimension	Metrik	Laufzeitverhalten
NNG	≥ 2	L_p	$O(n \cdot log(n))$
k-NNG	≥ 2	L_p	$O(kn \cdot log(n))$
MST	2	L_2	$O(n \cdot log(n))$
	> 2	L_2	$O(n^{2-2/(\lfloor d/2 \rfloor + 1)+\epsilon})$
RNG	2	L_2	$O(n \cdot log(n))$
		L_1, L_∞	$O(n \cdot log(n))$
	3	L_2	$O(n^{3/2+\epsilon})$
		L_1, L_∞	$O(n \cdot log^2(n))$
	> 3	L_2	$O(n^{2(1-\frac{1}{d+1})+\epsilon})$
		L_1, L_∞	$O(n \cdot log^{(d-1)}(n))$
GG	2	L_2	$O(n \cdot log(n))$
	> 2	L_p	$O(n^{\lfloor d/2 \rfloor})$
DT	2	L_2	$O(n \cdot log(n))$
	> 2	L_p	$O(n^{\lfloor d/2 \rfloor})$

Tabelle 6.2: Laufzeitverhalten der Berechnung von Nachbarschaftsgraphen.

Kapitel 7

Hierarchisches Nachbarschaftsgraphen Clustering

In diesem Kapitel wird unser Ansatz eines *parameterfreien* Clusterverfahrens zur Gruppierung räumlich verteilter *punkthafter* Objekte beschrieben. Unter punkthaft verstehen wir, dass die zu gruppierenden Objekte in einen n-dimensionalen Vektorraum abgebildet werden können. Am Ende des Kapitels werden wir noch darauf eingehen, wie das Verfahren auch auf metrische Objekträume, die nicht eindeutig auf einen Vektorraum abgebildet werden können, anwendbar ist.

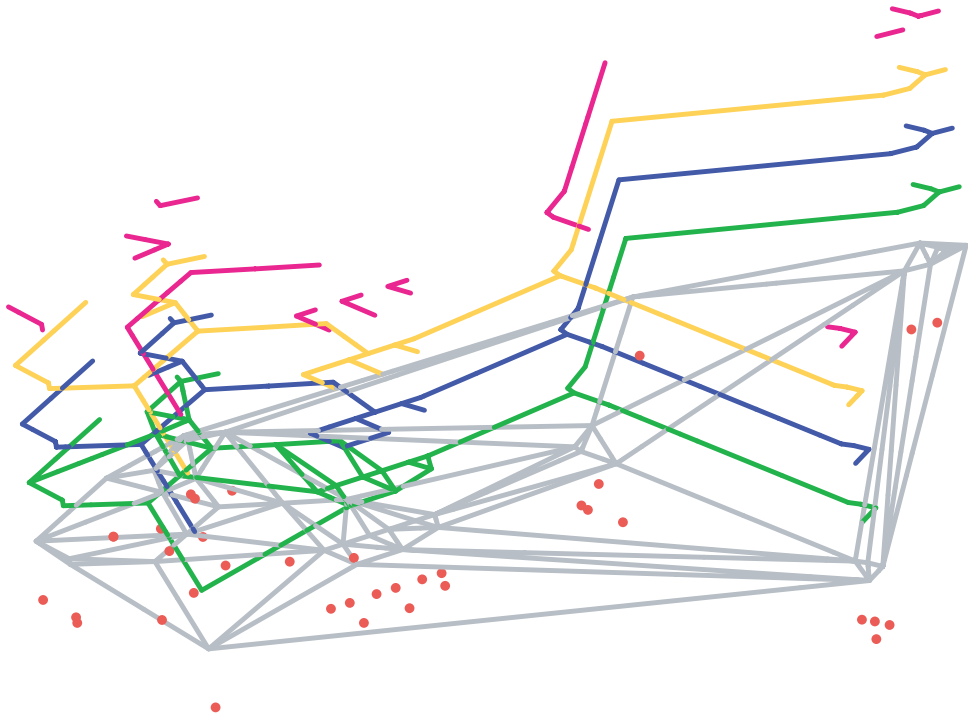


Abbildung 7.1: Nachbarschaftshierarchie

Unser Verfahren gehört zu den dichtebasierten, agglomerativen Graph-Clusterverfahren (siehe Kapitel 4). Die geeignete Verwendung der im Kapitel 6 beschriebenen Graphenhierarchie (Abb. 6.11, 7.1) und eine im folgenden beschriebene *medianbasierte Ähnlichkeitsrelation* ermöglicht es uns, Punktobjekte eines gegebenen Merkmalraums zu Gruppen zusammenzufassen, ohne dabei spezifisches Vorwissen, wie z.B. Verteilungsfunktionen und Schwellwerte als Entscheidungskriterium oder Puffergrößen zur Bestimmung der Nachbarschaft vorauszusetzen.

7.1 Warum Nachbarschaftsgraphen?

Wie schon in Kapitel 4 erwähnt, benötigt jedes Clusterverfahren eine Definition der Ähnlichkeit zwischen zwei Elementen aus der zu gruppierenden Objektmenge. Das Wort Element kann hier direkt als Synonym für Zahl, Attribut, Datum, etc. angesehen werden. Als ganz natürliche Definition für Ähnlichkeit zwischen zwei Elementen bietet sich dann eine *Metrik* an (siehe Kapitel 5, Def. 5.3.2). – eine Funktion also, die zwei Elemente der Objektmenge in den Raum der positiven reellen Zahlen inklusive der Null abbildet. Zwei Objekte sind nach dieser Definition dann *identisch*, wenn die Distanz zwischen beiden gleich Null ist und sie sind sich um so *unähnlicher*, je größer der Abstand zwischen beiden ist. Ein Clusterverfahren hat nun die Aufgabe zu lösen, die Menge der Objekte in Teilmengen *gleicher Ähnlichkeit* einzuteilen, wobei im allgemeinen davon auszugehen ist, dass es keine identischen Objekte gibt.

Die menschliche Wahrnehmung zeigt uns, dass wir Menschen automatisch Objekte in Gruppen einteilen, innerhalb derer ein homogener Abstand besteht und zusätzlich wirken Gruppen von Objekten mit geringem Abstand untereinander besonders *interessant* (Abbildung 7.2 und 4.1). Es gilt hier Toblers Gesetz „Everything is related to everything else, but near things are more related than distant things“ (Tobler 1970). Auf dieser Eigenschaft bauen nun die sogenannten dichtebasierten Clusterverfahren auf. Diese Verfahren versuchen, die Dichte von Objekten zu schätzen und Objekte gleicher Dichte zusammenzufassen.

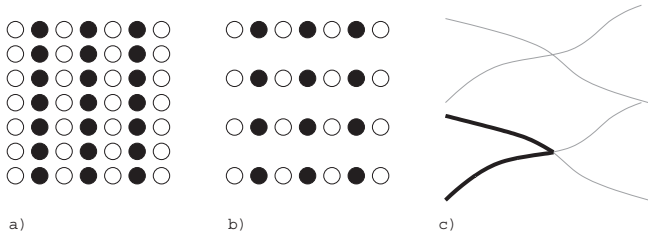


Abbildung 7.2: Gestalt-Gesetze: a) Gesetz der Ähnlichkeit, b) Gesetz der Nähe, c) Gesetz der guten Fortsetzung.

Physikalisch definiert sich die Dichte als Anzahl von Teilchen pro Volumeneinheit. Zur Schätzung der Dichte bieten sich nun mehrere Möglichkeiten an. In den meisten praktischen Problemfällen lassen sich die zu gruppierenden Objekte in einen n-dimensionalen Vektorraum abbilden. In solchen metrischen Räumen bietet es sich zum Beispiel an, den Objektraum in ein gleichmäßiges Zellraster mit vorgegebener Kantenlänge einzuteilen und die Anzahl der Objekte innerhalb einer Zelle zu bestimmen. Die Dichte ergibt sich dann aus dem Quotienten von Objektanzahl und Zellenfläche. Jedem Objekt in einer Zelle wird dann die entsprechende Dichte zugeordnet. Eine andere Möglichkeit ist es, anstatt eines Rasters einen festen Radius vorzugeben und um jeden Objektpunkt eine n-dimensionale Sphäre zu legen (Abb. 6.8), um dann die Anzahl der Objekte innerhalb der Sphäre zu bestimmen. Die Anzahl der Objekte ist dann ein Maß für die Dichte. Ester et al. (1996) führen auf dieser Basis den Begriff *density reachable* ein. Bei graphbasierten Verfahren bietet es sich zum Beispiel an, die Familie der k-NNG ((Jarvis & Patrick 1973), (Eppstein et al. 1997), (Karypis et al. 1999)) zu verwenden und die *mittlere Kantenlänge* aller inzidenter Kanten einer Ecke als deren Dichte zu definieren. Kantenlänge und Mittelwert müssen dabei natürlich in geeigneter Weise definiert werden. In den meisten Fällen wird die Distanz der zur Kante gehörenden Ecken verwendet, die wiederum von der gewählten Metrik abhängt. Als Mittelwert wird meistens das arithmetische Mittel verwendet.

All diese Methoden haben jedoch den Nachteil, dass sie mindestens einen Parameter benötigen (Zellengröße, Radius, Anzahl der zu betrachtenden Nachbarobjekte), um die Dichtebestimmung durchführen zu können. Nachbarschaftsgraphen haben nun, wie im Kapitel 6 beschrieben, die Eigenschaft, Nachbarschaft in einer eindeutigen mathematischen (natürlichen) Weise zu definieren. Wiederum kann der Begriff Nachbarschaft in Merkmalsräumen mit dem Begriff Ähnlichkeit gleichgesetzt werden. Wenn wir nun noch eine Metrik auf der Menge der Objekte des betrachteten Merkmalraums definieren können und diesen Distanzwert der Kante zwischen den beiden Ecken des zugeordneten Nachbarschaftsgraphen zuweisen, haben wir damit auch den Begriff der Dichte in Nachbarschaftsgraphen eingeführt.

Die Hierarchie der Nachbarschaftsgraphen wiederum definiert auf eine natürliche Weise eine Generalisierung des Begriffs Nachbarschaft und stellt uns somit eine Art *Lupeneffekt* zur Verfügung, den wir für einen verallgemeinerten *Buffergrowing* Algorithmus verwenden können. Die zugrunde liegende Idee ist, dass wir für eine robuste

Dichteschätzung erst alle am nächsten liegenden Nachbarn in Betracht ziehen und im weiteren sukzessive weiter entfernte Nachbarn hinzunehmen. In Standardverfahren würde man hierzu eine Pufferbreite definieren und diese Schritt für Schritt vergrößern. Bei diesem Vorgehen steht man jedoch vor dem Problem, eine Pufferform, sowie einen Startwert, einen Inkrementwert und einen Endwert für die Pufferbreite festlegen zu müssen. Die Hierarchie der Nachbarschaftsgraphen liefert uns diese Werte implizit. Die Form und Größe des Puffers wird durch die Nachbarschaftsbeziehungen im betrachteten Nachbarschaftsgraphen festgelegt. Der Startwert unseres Algorithmus wird der Nächste-Nachbar-Graph sein, da er die dichtesten Nachbarn umfasst und somit die feinste Auflösung bietet. Der Endwert wird durch die Delaunaytriangulation festgelegt, die die umfassendste topologische Nachbarschaftsbeziehung darstellt. Die schrittweise Vergrößerung der Pufferbreite wird durch die Auswahl des jeweils nächsten Nachbarschaftsgraphen (in der Reihenfolge ihrer Teilmengenbeziehung) in einer *natürlichen, topologischen* Weise festgelegt.

Im Kapitel 6 hatten wir beschrieben, dass zwischen den Nachbarschaftsgraphen Teilmengenbeziehungen (Gleichung 6.12, 6.13 und 6.14) existieren. Wir werden im Algorithmus die Teilmengenbeziehung 6.12 verwenden, obwohl der NNG im allgemeinen nicht immer eine Teilmenge des MST ist. Der NNG und der MST sind jedoch Teilmengen vom RNG und da der NNG im allgemeinen ein nicht zusammenhängender Graph ist und der MST, wie sein Name schon sagt, der minimal zusammenhängende Graph ist, stellen die vom MST eingefügten Kanten die kleinste Menge von Kanten dar, um zu einem zusammenhängenden Graphen zu gelangen. Wir sehen diesen Übergang als wichtig an, um benachbarte linienförmige Cluster, möglichst ohne Verlust ihrer linienhaften Form, zu gruppieren.

7.2 Was ist ein Nachbarschaftsgraphen-Cluster?

Bevor wir beginnen, unsere Definition von ähnlichen Clustern zu beschreiben, müssen wir erst definieren, was wir im folgenden unter einem Cluster in einem Nachbarschaftsgraphen verstehen und was die wesentlichen Eigenschaften solcher Cluster sind.

Wir definieren einen Cluster als Teilgraph eines Nachbarschaftsgraphen (Kap. 6, Def.6.1.4), wobei die Kantenmenge eines Clusters in zwei Mengen – die Menge der inneren Kanten und die Menge der äußeren Kanten – aufgeteilt werden kann. Innere Kanten verbinden zum Cluster gehörende Ecken und äußere Kanten verbinden jeweils eine Ecke des Clusters mit einer Ecke eines anderen Clusters. Die Menge der äußeren Kanten enthält die Information über die Nachbarschaft eines Clusters und die innere Kantenmenge enthält die Information über die Dichte eines Clusters. Da wir in unserem Modell einen Cluster als ein *unscharfes* Objekt (Fuzzy-Objekt) bzgl. seiner Abgrenzung zu Nachbarclustern ansehen, definieren wir im folgenden ein Maß für die Unschärfe der Clusterdichte anhand der inneren und äußeren Kantenmenge. Abbildung 7.3 veranschaulicht diesen Sachverhalt bildlich. In gleicher Weise kann man auch die Eckenmenge eines Clusters in eine innere und äußere Eckenmenge einteilen. Die äußere Eckenmenge stellt den Rand eines Clusters dar und kann zur Bestimmung der Form des Knoten-Clusters verwendet werden. Im weiteren wird aber auf die Einteilung der Eckenmenge nicht eingegangen, da für unser Verfahren diese Unterteilung nicht von Relevanz ist. Die folgenden Definitionen beschreiben unsere Clusterdefinition formal und führen die von uns verwendeten Notationen ein.

Definition 7.2.1 (Cluster)
Gegeben sei ein Nachbarschaftsgraph $\mathbf{G}(\mathbf{E}, \mathbf{K})$ *und eine Metrik* $\delta : E \times E \rightarrow R_0^+$*, dann definiert jeder Teilgraph* $\mathbf{C}(\mathbf{G})$ *einen Cluster dieses Graphen und* δ *ordnet jeder Kante eines Clusters einen Distanzwert (Länge) zu. Für die Eckenmenge eines Clusters schreiben wir im folgenden* $\mathbf{E}(\mathbf{C})$ *und für die Kantenmenge* $\mathbf{K}(\mathbf{C})$*. Eine Kante* $(\mathbf{e_i}, \mathbf{e_j})$ *bezeichnen wir im folgenden kurz mit* $\mathbf{k_{i,j}}$*.*

Wir werden im weiteren für $\mathbf{C}(\mathbf{G})$ nur $\mathbf{C}$ schreiben und den zugehörigen Graphen implizit als gegeben ansehen.

Definition 7.2.2 (Cluster-Intra-Kantenmenge)
Die $\mathbf{Intra-Kantenmenge}$ *– oder auch Innenkantenmenge genannt – sind alle Kanten, die Ecken des Clusters miteinander verbinden. Diese Kanten beschreiben die sogenannte* $\mathbf{Intra-Dichte}$ *eines Clustes, d.h. die Ähnlichkeit aller Ecken eines Clusters. Wir schreiben im folgenden für diese Intra-Kantenmenge* $\mathbf{C^o}$*, dabei gilt:*

$$\mathbf{C^o} = \{\mathbf{k_{ij}} \in \mathbf{K}(\mathbf{C}) \mid \mathbf{e_i} \in \mathbf{E}(\mathbf{C}) \ \textit{und} \ \mathbf{e_j} \in \mathbf{E}(\mathbf{C})\}$$

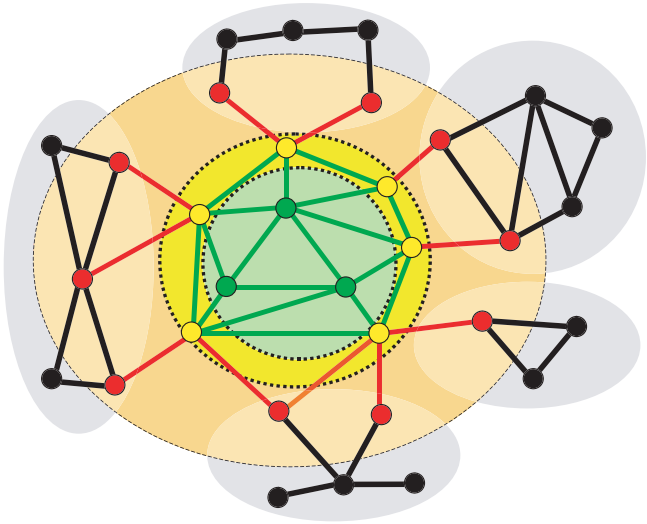


Abbildung 7.3: Clusterdefinition: Grün innere Kanten, rot äußere Kanten und gelb Clusterrand.

Definition 7.2.3 (Cluster-Inter-Kantenmenge)
Die **Inter − Kantenmenge** *– oder auch Außenkantenmenge genannt – sind alle Kanten, die Ecken verschiedener Cluster miteinander verbinden. Diese Kanten beschreiben die sogenannte* **Inter − Dichte** *zwischen Clustern, d.h. die Ähnlichkeit des Clusters zu allen anderen Clustern. Wir schreiben im folgenden für diese Inter-Kantenmenge* $\mathbf{C}^*$*, dabei gilt:*

$$\mathbf{C}^* = \{\mathbf{k_{ij}} \in \mathbf{K(C)} \mid (\mathbf{e_i} \in \mathbf{E(C)} \text{ und } \mathbf{e_j} \notin \mathbf{E(C)}) \text{ oder } (\mathbf{e_i} \notin \mathbf{E(C)} \text{ und } \mathbf{e_j} \in \mathbf{E(C)})\}$$

Definition 7.2.4 (Cluster-Rand)
Der Rand eines Clusters ist die Menge aller Ecken des Clusters, die zu einer Inter-Kante und zu einer Intra-Kante inzident sind. Wir schreiben im folgenden für den Rand eines Clusters $\partial\mathbf{C}$*.*

Definition 7.2.5 (Cluster-Komplement)
Das **Komplement** $\overline{\mathbf{C}}$ *eines Clusters* $\mathbf{C(G)}$ *definiert sich wie folgt:*

$$\overline{\mathbf{C}} = \{\mathbf{k_{i,j}} \mid \mathbf{k_{i,j}} \in (\mathbf{K(G)} \setminus \mathbf{K(C)})\}$$

Definition 7.2.6 (Regulärer Cluster)
Ein Cluster $\mathbf{C}$ *heißt* ***regulär****, wenn gilt:* $\mathbf{C^o} \neq \emptyset$*.*

Definition 7.2.7 (Abgeschlossener Cluster)
Ein Cluster $\mathbf{C}$ *heißt* ***abgeschlossen****, wenn gilt:* $\mathbf{C}^* = \emptyset$*.*

Definition 7.2.8 (Offener Cluster)
Ein nicht abgeschlossener Cluster $\mathbf{C}$ *wird als* ***offen*** *bezeichnet.*

Definition 7.2.9 (Singulärer Cluster)
Ein Cluster $\mathbf{C}$ *heißt* ***singulär*** *oder* ***Singularität****, wenn er abgeschlossen und nicht regulär ist.*

Definition 7.2.10 (Vereinigung zweier Cluster)
Die Vereinigung zweier Cluster $\mathbf{U} = \mathbf{X} \cup \mathbf{Y}$ *ist wie folgt definiert:*

$$\mathbf{E}(\mathbf{U}) = \mathbf{E}(\mathbf{X}) \cup \mathbf{E}(\mathbf{Y}) \;\; \textit{und} \;\; \mathbf{K}(\mathbf{U}) = \mathbf{K}(\mathbf{X}) \cup \mathbf{K}(\mathbf{Y})$$

7.3 Schätzung von Clustermerkmalen

Der Begriff *Dichte* stellt in unserem Verfahren die zentrale Rolle dar und bedarf deshalb einer genauen Definition. Wir gehen im weiteren davon aus, dass eine Metrik (Distanzfunktion) (vgl. Abschnitt 5.3) definiert ist, die den Abstand zwischen zwei Objekten der zu untersuchenden Objektmenge bestimmt[1]. Die Dichte eines Objekts steht dann im direkten Zusammenhang mit den Abständen zu seinen Nachbarobjekten. Es bietet sich daher an, die Dichte eines Objekts anhand der gegebenen Abstände zu schätzen. Es stellt sich uns nun die Frage, wie man anhand der Daten, ohne jegliches Vorwissen, eine Schätzung für die Dichte durchführen kann. Wir werden im folgenden nicht vertieft auf das Gebiet der angewandten Statistik und der robusten Schätzung sowie der sogenannten *parameterfreien* Statistik eingehen. Für eine Vertiefung empfehlen wir (Sachs 1999). Im folgenden werden nur kurz die Begriffe angesprochen, die zur Wahl unseres Modells geführt haben.

Aus der Statistik wissen wir, dass der Mittelwert und die Standardabweichung charakteristische Werte einer *symmetrischen Glockenkurve, Gaußschen Kurve* oder der *Normalverteilung* darstellen. Sie geben die Lage oder Lokalisation des durchschnittlichen oder mittleren Wertes einer Meßreihe und die Streuung oder Dispersion der Einzelwerte um den Mittelwert an. Darüber hinaus zeigt die Tschebyscheffsche Ungleichung 7.1, dass die Standardabweichung unabhängig von der Normalverteilung als allgemeines Streuungsmaß dienen kann. Dies ist für die Clusteranalyse von wesentlicher Bedeutung, da man im allgemeinen nicht von einer Normalverteilung ausgehen kann. Insbesondere bei raumbezogenen (geographischen) Daten haben wir es häufig mit nicht normalverteilten Daten zu tun. Raumbezogene Daten lassen sich besser als Menge von Objektmengen charakterisieren, wobei jede dieser Teilmengen durch eine eigene Normalverteilung bzgl. der Objektabstände innerhalb der Teilmenge charakterisiert werden kann.

$$\text{Für beliebige Verteilungen gilt:} \quad P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2} \quad \text{mit} \quad k > 0 \tag{7.1}$$

Das Arithmetische Mittel 7.2 und die Standardabweichung 7.3 haben jedoch wesentliche Nachteile, wenn keine Information über die Verteilung der Daten und über etwaige Ausreißer (Extremwerte) bekannt ist. Das arithmetische Mittel ist als Schätzwert für den Erwartungswert einer Verteilung nur geeignet, wenn die Verteilung eingipflig und angenähert symmetrisch ist. Das arithmetische Mittel ist um so weniger geeignet, je schiefer die Verteilung und je größer die Streuung ist. Die Varianz bzw. die Standardabweichung ist um so weniger brauchbar, je stärker die Beobachtungen sich voneinander unterscheiden. Extremwerte verzerren Schätzwerte extrem – insbesondere dann, wenn zu ihrer Berechnung die Summe der *Abweichungsquadrate* der Einzelwerte vom Mittelwert benötigt wird, d.h. wenn die Varianz oder die Standardabweichung, der Korrelationskoeffizient oder ein Regressionskoeffizient zu schätzen sind. In den genannten Fällen liefert der Zentralwert oder **Median** ($\tilde{x}$) bessere Ergebnisse[2]. Der Median gibt den beobachteten Wert an, der die Verteilung einer Stichprobe in zwei gleich große Hälften teilt.

$$\textbf{Arithmetisches Mittel:} \quad \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{7.2}$$

$$\textbf{Empirische Standardabweichung:} \quad s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{7.3}$$

[1] Wenn wir im weiteren Verlauf von Objekten reden, sind damit die Daten gemeint, die einer Ecke des augenblicklich betrachteten Nachbarschaftsgraphen zugeordnet sind.

[2] Ein statistisches Verfahren (Schätzfunktion oder Test), das unempfindlich ist gegenüber verunreinigten Daten (z.B. durch Ausreisser), wird *resistant* (widerstandsfähig) genannt. Das arithmetische Mittel und die Standardabweichung sind äußerst *nonresistant*. Der Median wird dagegen erst bei 50% Verunreinigung verzerrt (Sachs 1999).

Der Median kann nur für ordinalskalierte Mengen (Stichproben) bestimmt werden, d.h. es muss – durch eine $\leq$ oder $\geq$ Relation – für die Elemente der Stichprobe eine Rangordnung definiert sein, so dass die Stichprobe als auf- oder absteigende Liste interpretiert werden kann.

Definition 7.3.1 (Median)
Sei $X = \{x_1, x_2, x_3, \ldots, x_n\}$ eine n-elementige Menge (Stichprobe), die auf- oder absteigend angeordnet ist, d.h. es gilt entweder $\{x_1 \leq x_2 \leq x_3 \leq \cdots \leq x_n\}$ oder $\{x_1 \geq x_2 \geq x_3 \geq \cdots \geq x_n\}$, dann definiert sich der $\mathbf{Median}(\mathbf{X})$ *wie folgt:*

$$\mathbf{Median}(\mathbf{X}) = \begin{cases} x_{\lceil n/2 \rceil}, & \text{falls } n \text{ ungerade} \\ \frac{1}{2}(x_{n/2} + x_{n/2+1}), & \text{sonst} \end{cases}$$

Im Falle von geradzahligen Stichproben spricht man auch besser vom *Pseudomedian*. Ist eine Stichprobe in Klassen eingeteilt, dann kann der Pseudomedian durch lineare Interpolation bestimmt werden (siehe (Sachs 1999) Gleichung 1.102). Der Pseudomedian kann nur im Falle von quantitativen Daten erfasst werden. Im Falle von rein ordinalen Werten muss man sich im allgemeinen für $x_{n/2}$ oder $x_{n/2+1}$ entscheiden, da der Wert $0.5(x_{n/2}+x_{n/2+1})$ nicht definiert sein kann. Wir werden im folgenden nicht das arithmetische Mittel sondern den Median verwenden, da er im Gegensatz zum arithmetischen Mittel **robust** ist. Der Median ist beispielsweise, im Gegensatz zum arithmetischen Mittel, unempfindlich gegenüber der Modellvoraussetzung *Normalverteilung* und ist in den folgenden Fällen zu bevorzugen:

- Kleine Stichprobenumfänge,
- asymmetrische Verteilungen,
- Verteilungen mit offenen Endklassen,
- Verdacht auf Ausreißer und
- bei ordinalskalierten Stichproben (Rangdaten).

Wenn wir im weiteren von Mengen oder Stichproben reden, meinen wir ab jetzt immer implizit geordnete Mengen, d.h. z.B. dass die Kantenmengen unserer Cluster aufsteigend nach der Länge der Kanten angeordnet sind.

Definition 7.3.2 (Cluster-Dichte)
Als Schätzwert für die Dichte eines Clusters $\mathbf{X}$ *wählen wir den Median der inneren Kantenlängen:*

$$\mathbf{D}(\mathbf{X}) = \mathbf{Median}(\{\mathbf{d_{i,j}} \mid \mathbf{k_{i,j}} \in \mathbf{X^o}\})$$

Als Maß für die Dichte eines Clusters verwenden wir also die robuste Schätzung der erwarteten Kantenlänge für die inneren Kanten des Clusters. Je größer der Erwartungswert der inneren Kantenlänge, desto geringer ist die Dichte des Clusters. Die alleinige Verwendung der Dichte zum Vergleich von benachbarten Clustern auf Gleichheit oder – besser gesagt – auf Zusammengehörigkeit kann dann nur durch Gleichheit beider Dichten bestätigt werden. Jedes Cluster ist jedoch eine Schätzung für eine zusammenhängende Kantenmenge mit einer bestimmten erwarteten Kantenlänge und diese Kanten werden nur in den seltensten Fällen exakt gleiche Kantenlängen besitzen und somit würde das Clusterverfahren praktisch keine Cluster finden können. Aus diesem Grund muß eine Unschärfe für die Dichte eines Clusters definiert werden. Dies wird üblicherweise durch ein Toleranzintervall modelliert. Die Bestimmung dieses Toleranzintervalls kann entweder durch einen Schwellert definiert werden (z.B. $D(X_i) - D(X_j) < t$) oder durch eine weitere Schätzung für die Varianz der Kantenlängen eines Clusters.

Die Unschärfe eines Clusters (*Cluster-Toleranz*) baut in unserem Verfahren auf der Definition der *Cluster-Varianz* als Erwartungswert der Abweichung der Kantenlänge aller Kanten eines Clusters (innere und äußere Kanten) vom Erwartungswert der inneren Kantenlänge auf. Für die robuste Schätzung der Varianz verwenden wir den *Median aller absoluten Abweichungen vom Median* einer Stichprobe, welcher mit **MAD** (Median Absolute Deviation) bezeichnet wird und robuster ist als der Mittelwert der absoluten Abweichungen vom Mittelwert, die häufig auch mit MAD abgekürzt wird, was leicht zu Verwechslungen führen kann.

Definition 7.3.3 (Cluster-Varianz)
Den Schätzwert für die Varianz der zu erwartenden Kantenlänge eines Cluster $\mathbf{X}$ *definieren wir wie folgt:*

$$\mathbf{V}(\mathbf{X}) = \mathbf{MAD}(\mathbf{X}) = \mathbf{Median}(\{|\delta(\mathbf{k_{i,j}}) - \mathbf{D}(\mathbf{X})| \mid \mathbf{k_{i,j}} \in \mathbf{E}(\mathbf{X})\})$$

Definition 7.3.4 (Cluster-Toleranz)
Das Toleranzintervall für die Dichte eines Clusters $\mathbf{X}$ *ergibt sich aus den Definitionen 7.3.2 und 7.3.3 wie folgt:*

$$\mathbf{T}(\mathbf{X}) = [\mathbf{Min}(\mathbf{X}), \mathbf{Max}(\mathbf{X})]$$

$$\textit{mit} \quad \mathbf{Min}(\mathbf{X}) = \mathbf{D}(\mathbf{X}) - \mathbf{V}(\mathbf{X}) \;\; \textit{und} \;\; \mathbf{Max}(\mathbf{X}) = \mathbf{D}(\mathbf{X}) + \mathbf{V}(\mathbf{X})$$

Falls $\mathbf{Min}(\mathbf{X}) < \mathbf{0}$ *setzen wir* $\mathbf{Min}(\mathbf{X}) = \mathbf{0}$*, da keine negativen Abstände (Dichten) existieren können.*

Diese Modellierung der Cluster-Dichte und Cluster-Varianz ermöglicht uns eine einheitliche Schätzung der Clusterdichte für reguläre und nicht reguläre Cluster. Was uns jetzt noch fehlt, ist die Definition des Abstandes zweier benachbarter Cluster X und Y. Diese definieren wir analog zur Cluster-Dichte und Cluster-Varianz auf der Schnittmenge der äußeren Kantenmengen von X und Y.

Definition 7.3.5 (Cluster-Abstand)
Der Abstand zwischen zwei Clustern $\mathbf{X}$ *und* $\mathbf{Y}$ *wird durch folgendes Intervall definiert:*

$$\mathbf{\Delta}(\mathbf{X}, \mathbf{Y}) = [\mathbf{MD}(\mathbf{X}, \mathbf{Y}) - \mathbf{VD}(\mathbf{X}, \mathbf{Y}), \mathbf{MD}(\mathbf{X}, \mathbf{Y}) + \mathbf{VD}(\mathbf{X}, \mathbf{Y})]$$

$$\textit{mit} \quad \mathbf{MD}(\mathbf{X}, \mathbf{Y}) = \mathbf{Median}(\mathbf{d_{i,j}} \mid \mathbf{k_{i,j}} \in \mathbf{X}^* \cap \mathbf{Y}^*)$$

$$\textit{und} \quad \mathbf{VD}(\mathbf{X}, \mathbf{Y}) = \mathbf{Median}(\{|\delta(\mathbf{k_{i,j}}) - \mathbf{MD}(\mathbf{X}, \mathbf{Y})| \mid \mathbf{k_{i,j}} \in \mathbf{X}^* \cap \mathbf{Y}^*\})$$

Falls $\mathbf{MD}(\mathbf{X}, \mathbf{Y}) - \mathbf{VD}(\mathbf{X}, \mathbf{Y}) < \mathbf{0}$ *setzen wir* $\mathbf{MD}(\mathbf{X}, \mathbf{Y}) - \mathbf{VD}(\mathbf{X}, \mathbf{Y}) = \mathbf{0}$*, da keine negativen Abstände existieren können.*

7.4 Medianbasierte Ähnlichkeitsrelation zweier Cluster

Aufbauend auf den Definitionen der Cluster-Dichte, Cluster-Varianz und des Cluster-Abstands können wir nun einfache parameterfreie Vergleichskriterien definieren. Als erstes definieren wir eine Bedingung, die notwendig ist, um Cluster gleicher Dichte voneinander zu trennen, die zu weit entfernt voneinander sind.

Definition 7.4.1 (Abstands-Kompatibilität)
Zwei Cluster $\mathbf{X}$ *und* $\mathbf{Y}$ *werden als verträglich (kompatibel) bezüglich ihres Abstands betrachtet, wenn folgende Bedingung erfüllt ist:*

$$\mathbf{\Delta}(\mathbf{X}, \mathbf{Y}) \subseteq \mathbf{T}(\mathbf{X}) \;\; \textit{und} \;\; \mathbf{\Delta}(\mathbf{X}, \mathbf{Y}) \subseteq \mathbf{T}(\mathbf{Y})$$

Als zweites definieren wir eine Bedingung zum robusten Testen zweier Clusterdichten auf Ähnlichkeit (Quasi-Gleichheit).

Definition 7.4.2 (Intra-Dichte-Kompatibilität)
Zwei Cluster $\mathbf{X}$ *und* $\mathbf{Y}$ *werden als verträglich (kompatibel) bezüglich ihrer Dichte betrachtet, wenn folgende Bedingung erfüllt ist:*

$$\mathbf{D}(\mathbf{X}) \subseteq \mathbf{T}(\mathbf{Y}) \;\; \textit{und} \;\; \mathbf{D}(\mathbf{Y}) \subseteq \mathbf{T}(\mathbf{X})$$

Die Definition besagt, dass zwei Clusterdichten dann als gleich betrachtet werden, wenn ihre Erwartungswerte im Toleranzbereich der jeweils anderen Clusterdichte liegen (Abb. 7.4).

Wie man sieht, berechnen wir für die Kompatibilitätstests kein explizites Ähnlichkeitsmaß (oder Distanzmaß), jedoch lässt sich einfach zeigen, dass Definition 7.4.2 zu einem Ähnlichkeitsmaß in Beziehung steht, wenn man Definition 7.4.2 durch die äquivalente Bedingung

$$\frac{|\mathbf{D}(\mathbf{X}) - \mathbf{D}(\mathbf{Y})|}{\min\{\mathbf{V}(\mathbf{X}), \mathbf{V}(\mathbf{Y})\}} \leq \mathbf{1} \tag{7.4}$$

ersetzt, aus der sich dann ein Ähnlichkeitsmaß, wie folgt definieren lässt:

$$\mathbf{s}(\mathbf{X},\mathbf{Y}) = \begin{cases} 1 - \frac{|\mathbf{D}(\mathbf{X})-\mathbf{D}(\mathbf{Y})|}{\min\{\mathbf{V}(\mathbf{X}),\mathbf{V}(\mathbf{Y})\}} & \text{falls } |\mathbf{D}(\mathbf{X}) - \mathbf{D}(\mathbf{Y})| \leq \min\{\mathbf{V}(\mathbf{X}), \mathbf{V}(\mathbf{Y})\} \\ 0 & \text{sonst} \end{cases} . \tag{7.5}$$

In gleicher Weise ist es auch möglich, die Abstands-Kompatibilität, nach Definition 7.4.1, durch ein geeignetes Ähnlichkeitsmaß zu ersetzen, wie z.B. durch die Ähnlichkeitsfunktion

$$\mathbf{s}(\mathbf{X},\mathbf{Y}) = \frac{\mathbf{1}}{\mathbf{3}}(\mathbf{s}_{\mathbf{MD_X}}(\mathbf{X},\mathbf{Y}) + \mathbf{s}_{\mathbf{MD_Y}}(\mathbf{X},\mathbf{Y}) + \mathbf{s}_{\mathbf{VD}}(\mathbf{X},\mathbf{Y})) \quad \text{mit} \tag{7.6}$$

$$\mathbf{s}_{\mathbf{MD_X}}(\mathbf{X},\mathbf{Y}) = \begin{cases} 1 - \frac{|\mathbf{MD}(\mathbf{X},\mathbf{Y})-\mathbf{D}(\mathbf{X})|}{\mathbf{V}(\mathbf{X})} & \text{falls } |\mathbf{MD}(\mathbf{X},\mathbf{Y}) - \mathbf{D}(\mathbf{X})| \leq \mathbf{V}(\mathbf{X}) \\ 0 & \text{sonst} \end{cases} \quad \text{und} \tag{7.7}$$

$$\mathbf{s}_{\mathbf{MD_Y}}(\mathbf{X},\mathbf{Y}) = \begin{cases} 1 - \frac{|\mathbf{MD}(\mathbf{X},\mathbf{Y})-\mathbf{D}(\mathbf{Y})|}{\mathbf{V}(\mathbf{Y})} & \text{falls } |\mathbf{MD}(\mathbf{X},\mathbf{Y}) - \mathbf{D}(\mathbf{Y})| \leq \mathbf{V}(\mathbf{Y}) \\ 0 & \text{sonst} \end{cases} \quad \text{und} \tag{7.8}$$

$$\mathbf{s}_{\mathbf{VD}}(\mathbf{X},\mathbf{Y}) = \begin{cases} 1 - \frac{|\mathbf{\Delta}(\mathbf{X},\mathbf{Y})|}{\min\{|\mathbf{T}(\mathbf{X})|,|\mathbf{T}(\mathbf{Y})|\}} & \text{falls } |\mathbf{\Delta}(\mathbf{X},\mathbf{Y})| \leq \min\{|\mathbf{T}(\mathbf{X})|, |\mathbf{T}(\mathbf{Y})|\} \\ 0 & \text{sonst} \end{cases} . \tag{7.9}$$

Da der Erwartungswert für die Kantenlänge eines nicht regulären Clusters 0 ist und reguläre Cluster einen Erwartungswert ungleich 0 haben und im allgemeinen einen Toleranzbereich, der nicht die 0 enthält, können im allgemeinen Fall diese Cluster nach der strengen Definition mit regulären Clustern nicht vereinigt werden. Insbesondere bei regelmäßigen Strukturen kann es jedoch durch die in unserem Verfahren implizit definierten Abarbeitungsreihenfolge, die durch die Anordnung der Cluster in der Prioritätswarteschlange entsteht, dazu führen, dass nicht reguläre Cluster von regulären Clustern eingeschlossen sind und somit niemals gruppiert werden. Wir bezeichnen Cluster, die auf diese Weise entstehen, als *Sackgassen-Cluster*. In Abbildung 7.5 werden exemplarisch 4 mögliche Konstellationen von 1-, 2-, 3- oder 4-fach Sackgassen-Cluster in einem regelmäßigen 3×3 Gitter dargestellt. Selbstverständlich sind noch mehr Konstellationen möglich und es ist auch nicht zwingend eine regelmäßige Struktur nötig, um Sackgassen-Cluster zu erzeugen. Für nicht reguläre Cluster müssen wir deshalb Definition 7.4.2 wie folgt abschwächen:

Definition 7.4.3 (Intra-Dichte-Kompatibilität bei nicht regulären Clustern)
Zwei Cluster $\mathbf{X}$ *und* $\mathbf{Y}$*, von denen mindestens einer nicht regulär ist, werden dann als verträglich (kompatibel) bezüglich ihrer Dichte betrachtet, wenn folgende Bedingung erfüllt ist:*

$$\mathbf{D}(\mathbf{X}) \subseteq \mathbf{T}(\mathbf{Y}) \;\; \textit{oder} \;\; \mathbf{D}(\mathbf{Y}) \subseteq \mathbf{T}(\mathbf{X})$$

Bei nicht regulären Clustern genügt es also schon, wenn einer der Erwartungswerte im Toleranzbereich des anderen Clusters liegt. Das entspricht dem Fall V in Abbildung 7.4.

Was uns noch fehlt, ist ein geeignetes Homogenitätsmaß. Unser Ziel ist es, möglichst homogene Cluster zu bilden, d.h. also Cluster mit geringer Varianz. Deshalb fordern wir, dass die Varianz der Dichte des vereinigten Clusters mindestens kleiner ist als die Varianz eines der beiden einzelnen Cluster.

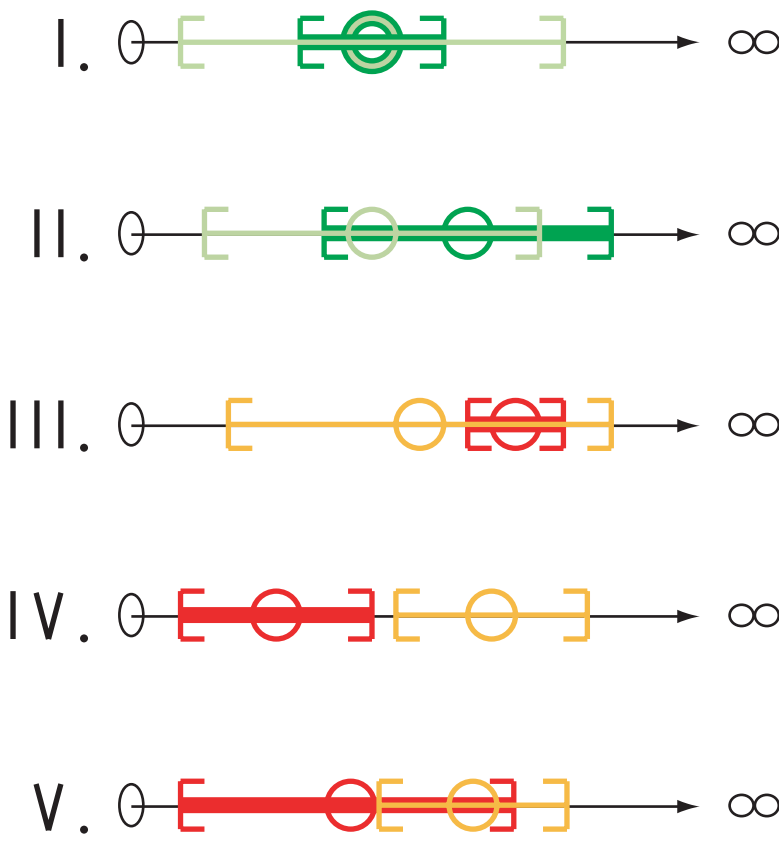


Abbildung 7.4: Intra-Dichte-Kompatibilität: Erwartungswerte (Kreise) und zugehörige Toleranzintervalle besitzen jeweils die gleiche Farbe. Die Intervalle in den Fällen I und II sind kompatibel. Die Intervalle in den Fällen III, IV und V sind dagegen nicht kompatibel. Im Falle eines nicht regulären Clusters gilt der Fall V dagegen als kompatibel (siehe Text).

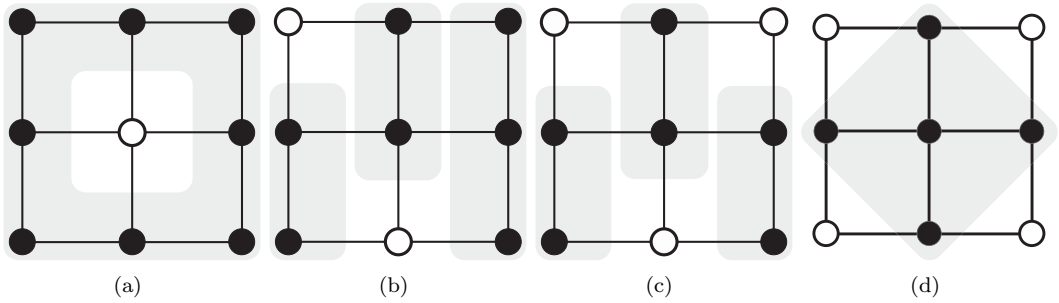


Abbildung 7.5: Beispiele für Sackgassen-Cluster (weiß)

Definition 7.4.4 (Varianz-Kompatibilität)
Zwei Cluster $\mathbf{X}$ *und* $\mathbf{Y}$ *werden als verträglich (kompatibel) bezüglich ihrer Varianz (Homogenität) betrachtet, wenn folgende Bedingung erfüllt ist:*

$$\mathbf{V}(\mathbf{X} \cup \mathbf{Y}) \leq \max\{\mathbf{V}(\mathbf{X}), \mathbf{V}(\mathbf{Y})\}$$

Eine strenge Definition einer optimalen Zerlegung einer Menge in homogene Cluster verlangt auch noch, dass der Unterschied zwischen den einzelnen Clustern so groß wie möglich wird. Deshalb führen wir als weiteres zum strengen Ähnlichkeitsvergleich die folgende Definition ein:

Definition 7.4.5 (Inter-Dichte-Kompatibilität)
Zwei Cluster $\mathbf{X}$ *und* $\mathbf{Y}$ *werden als verträglich (kompatibel) bezüglich ihrer Inter-Dichte (Abstand zu anderen Clustern) betrachtet, wenn folgende Bedingung erfüllt ist:*

$$\mathbf{D}(\mathbf{X}^*) \leq \mathbf{D}((\mathbf{X} \cup \mathbf{Y})^*) \;\; \textit{und} \;\; \mathbf{D}(\mathbf{Y}^*) \leq \mathbf{D}((\mathbf{X} \cup \mathbf{Y})^*)$$

7.5 HPGCL-Algorithmus

HPGCL steht für **H**ierarchisches **p**arameterfreies **G**raph-**Cl**ustering. Der HPGCL-Algorithmus besteht aus zwei Prozeduren. Die erste Prozedur erzeugt die Nachbarschaftsgraphenhierarchie und die zweite Prozedur führt die Gruppierung (Clustering) der Ecken der Nachbarschaftsgraphenhierarchie durch.

Die Nachbarschaftsgraphenhierarchie wird gemäß Definition 6.12 gebildet. Da die Delaunay Triangulation (DT) die Gesamtmenge aller Kanten darstellt, wird sie zuerst berechnet oder als gegeben vorausgesetzt. Danach werden alle Untergraphen in der folgenden Reihenfolge berechnet: Gabriel Graph (GG), Relativer Nachbarschaftsgraph (RNG), Minimaler spannender Baum (MST) und zuletzt der Nächste-Nachbar-Graph (NNG). Da sich alle Graphen in linearer Zeit aus ihren Obergraphen bestimmen lassen, besitzt diese Prozedur unseres Algorithmus die Komplexität der zugehörigen DT (siehe Tabelle 6.2 auf Seite 69). Nachdem die Hierarchie erzeugt wurde, existiert nun die DT und für jede Kante der DT ist gespeichert, zu welchem Untergraphen sie auch noch gehört.

Die Erzeugung der Nachbarschaftshierarchie kann als ein Bottom-Up Prozess angesehen werden, da wir mit der Delaunay-Triangulation beginnen und diese die Grundmenge aller möglichen Nachbarschaftsgraphkanten darstellt. Die Gruppierung der Knoten erfolgt dagegen in einem Top-Down Prozess vom NNG bis zur DT. Im folgenden wird nun der Grundalgorithmus unseres Clusterverfahrens und seine iterative Erweiterung beschrieben. Auf die Berechnung der Nachbarschaftsgraphen gehen wir hier nicht ein und verweisen auf die in Kapitel 6 angegebene Literatur.

7.5.1 Grundalgorithmus

Der Kern unseres HPGCL-Verfahrens kann wie folgt beschrieben werden:

Schritt 1: Zuerst werden die Kanten des NNG als *aktiv*[3] markiert. Danach werden alle Ecken der DT als *nicht reguläre Cluster* (Def. 7.2.6) initialisiert und in einer *Cluster-Prioritätswarteschlange* ($\mathcal{CPQ}$) gespeichert. Die *Cluster-Priorität* wird nach folgenden Kriterien bestimmt:

1. **Cluster-Varianz:** Ein Cluster hat umso größere Priorität, je homogener die Länge seiner inneren und äußeren Kanten ist (siehe Def. 7.3.3).
2. **Cluster-Abstand:** Ein Cluster hat umso größere Priorität, je geringer sein äußerer Abstand zu anderen Clustern ist (siehe Def. 7.3.5).

[3] Alle Berechnungen innerhalb des Gruppierungsprozesses (Clusterings) werden nur auf als aktiv markierte Kanten durchgeführt. Nicht markierte Kanten bleiben unberücksichtigt.

Diese Rangordnung wurde empirisch anhand der Testdaten ermittelt. Sie lässt sich doch auch logisch begründen. Da wir nach gleichmäßig strukturierten Clustern (der Unterschied zwischen den inneren Abständen eines Clusters sollte so gering wie möglich sein) suchen, bietet es sich für eine heuristische Suche nach der optimalen Zerlegung an, zuerst die besten Teillösungen zu betrachten (also die Cluster mit der geringsten Varianz); das zweite Kriterium folgt direkt aus dem allgemeinen *Nächsten-Nachbar-Prinzip*, d.h. gibt es Cluster mit gleicher Varianz, dann ist der Cluster zu bevorzugen, der sich am geringsten von seinen Nachbarclustern unterscheidet.

Setze den Index $\mathcal{I}_{\mathcal{CPQ}} = 0$, d.h. der Cluster mit der höchsten Priorität. Der $\mathcal{I}_{\mathcal{CPQ}}$ verweist auf den Cluster mit der *i-ten* Priorität und liegt im Intervall $[0, n]$, wobei n die Länge von $\mathcal{CPQ}$ ist.

Schritt 2: Wenn $\mathcal{I}_{\mathcal{CPQ}} = n$ ist, konnten keine Cluster gruppiert werden und wir fahren mit Schritt 6 fort, ansonsten wähle aus der $\mathcal{CPQ}$ den Cluster an der Stelle $\mathcal{I}_{\mathcal{CPQ}}$ aus. Wir bezeichnen diesen Cluster im folgenden mit **c**.

Schritt 3: Bestimme die Liste aller Nachbarcluster $\mathcal{N}$ von **c** mit Hilfe von $\overline{\mathbf{c}}$. Ist $\mathcal{N} = \emptyset$, d.h. es existieren keine Nachbarn, erhöhe $\mathcal{I}_{\mathcal{CPQ}}$ um 1 und fahre mit Schritt 2 fort.

Schritt 4: Entferne alle Cluster aus $\mathcal{N}$, die nicht kompatibel zu **c** sind. Die Kompatibilität[4] bestimmt sich dabei nach den folgenden Kriterien:

1. **Abstands-Kompatibilität**, gemäß 7.4.1
2. **Intra-Dichte-Kompatibilität**, gemäß 7.4.2
3. **Varianz-Kompatibilität**, gemäß 7.4.4
4. *optional* **Inter-Dichte-Kompatibilität**, gemäß 7.4.5

Schritt 5: Ist $\mathcal{N} = \emptyset$ erhöhe den $\mathcal{I}_{\mathcal{CPQ}}$ um 1 und fahre mit Schritt 2 fort. Ist $\mathcal{N} \neq \emptyset$ entferne **c** und alle Cluster in $\mathcal{N}$ aus $\mathcal{CPQ}$. Vereinige, nach Definition 7.2.10, **c** und alle Cluster in $\mathcal{N}$ zu einem neuen Cluster **m** und füge diesen neuen Cluster in $\mathcal{CPQ}$ ein. Setze $\mathcal{N} = \emptyset$, $\mathcal{I}_{\mathcal{CPQ}} = 0$ und fahre mit Schritt 2 fort[5].

Schritt 6: Auf der aktuellen Hierarchiestufe konnten keine kompatiblen Cluster mehr gefunden werden und $\mathcal{CPQ}$ enthält das Ergebnis dieses Gruppierungsschrittes. Diese Cluster werden nun in einer Clusterliste $\mathcal{CL}$ gespeichert und $\mathcal{CPQ} = \emptyset$ gesetzt.

Schritt 7: Sind wir noch nicht auf der Hierarchiestufe DT, dann aktivieren wir die Kanten der nächsten Hierarchiestufe und aktualisieren alle Cluster in $\mathcal{CL}$, da möglicherweise durch den Hierarchiewechsel neue Kanten zu den Clustern dazugekommen sind und sich somit die Dichte und Varianz der Cluster geändert hat. Nachdem alle Cluster aus $\mathcal{CL}$ aktualisiert worden sind, werden sie in die $\mathcal{CPQ}$ kopiert, $\mathcal{CL} = \emptyset$ und $\mathcal{I}_{\mathcal{CPQ}} = 0$ gesetzt und es wird mit Schritt 2 fortgefahren.

Schritt 8: Ende des Algorithmus erreicht.

7.5.2 Iterative Erweiterung

Wie bei jedem hierarchischen Clusterverfahren, kann man nun für die gefundenen Cluster eine neue Ähnlichkeitsmatrix oder einen neuen Nachbarschaftsgraphen berechnen und das Verfahren wiederholen. Dieses Vorgehen lässt sich solange wiederholen, bis nur noch ein einziger Cluster, die Gesamtmenge, übrig bleibt. Wir ändern dieses Vorgehen dahingehend ab, dass wir den Nachbarschaftsgraphen nicht wieder neu berechnen. Wir sagen, dass die globalen Nachbarschaftsverhältnisse und somit implizit die Ähnlichkeit unverändert bleibt und ersetzen jeden gefundenen Cluster durch einen einzelnen Knoten. Dazu werden alle inneren Kanten jedes einzelnen Clusters entfernt und da die Ecken ihre Clusterzugehörigkeit behalten, erhalten wir dadurch implizit einen neuen Graphen, der nur noch die Kanten aller äußeren Kantenmengen enthält und in dem die Ecken durch die gefundenen Cluster repräsentiert werden. Das Gruppieren der Cluster erfolgt dann in gleicher Weise wie oben beschrieben. Dieses Vorgehen erspart uns die Neuberechnung des Graphen und wir werden im allgemeinen eine

[4] Da es während des Verfahrens mehrfach zum Vergleich der gleichen Cluster kommen kann, merken wir uns in einer Tabelle die Cluster Id's und das Ergebnis des Vergleichs und überprüfen immer erst, ob ein Vergleichsergebnis schon existiert. Werden zwei Cluster miteinander vereinigt, so werden alle Tabelleneinträge mit diesen Id's entfernt.

[5] Optional erlaubt unser Verfahren auch nur die Vereinigung mit dem *nächsten* kompatiblen Cluster (Cluster mit dem kleinsten Abstand).

Grenzzerlegung der Ausgangsmenge finden, die ungleich der Gesamtmenge ist. Das ist deshalb der Fall, da die Menge der äußeren Kanten sukzessive immer kleiner wird und im allgemeinen, bei unregelmäßig verteilten Objekten, abgeschlossene Cluster entstehen, die nicht mehr vereinigt werden können. Der Fall eines einzigen Clusters als Ergebnis ist somit nur noch die Ausnahme, und die Vorgabe eines Parameters als Abbruchkriterium wird somit vermieden.

Wir ersetzen deshalb Schritt 8 des Algorithmus durch folgende Anweisungen:

Schritt 8: Hat sich die Anzahl der Cluster seit der letzten Iteration verringert, entfernen wir die inneren Kanten der Cluster aus der Nachbarschaftshierarchie[6] und fahren mit Schritt 1 fort, nur dass jetzt nicht die einzelnen Ecken der Nachbarschaftsgraphenhierarchie als nicht-reguläre Cluster initialisiert werden, sondern die gefundenen Ecken-Cluster der vorangegangenen Iteration.

Schritt 9: Die Anzahl der Cluster hat sich seit der letzten Iteration nicht verringert und das Ende des Algorithmus ist erreicht.

Unser Verfahren kann also wirklich als gänzlich parameterfrei betrachtet werden und ist somit hervorragend zur explorativen Analyse unbekannter raumbezogener Daten geeignet.

7.6 Verallgemeinerungen des Verfahrens

Bisher sind wir davon ausgegangen, dass unser zu zerlegender Objektraum auf einen n-dimensionalen Vektorraum abgebildet werden kann. Das ist z.B. für Multispektraldaten, Laserdaten oder räumliche Objekte, deren Lage im Raum durch Punkte beschrieben werden kann, immer möglich. Im allgemeinen haben geographische Objekte jedoch eine komplexe Form, dargestellt durch Punkte, Linien (Kurven), Flächen oder sogar Volumen (z.B. in Form eines Drahtmodells, einer Randbeschreibung oder eines CSG-Modells), und besitzen nicht nur quantitative sondern auch qualitative Attribute (siehe Kapitel 5). Qualitative Attribute können z.B. Straßennamen, Nutzungsarten, Gebäudetypen oder Kartensignaturen sein. Im folgenden werden wir beschreiben, wie sich solche Problemfälle auf unser oben beschriebenes Verfahren abbilden lassen.

7.7 Verallgemeinerung auf polyederförmige Objekte

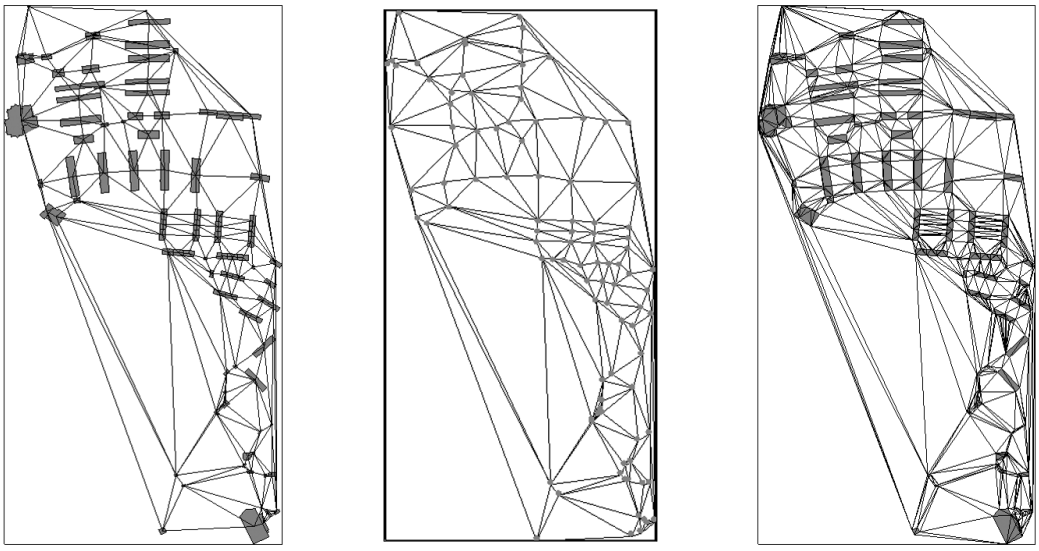
Im folgenden beschreiben wir, wie wir in unserem Verfahren Polygone und Polyeder behandeln können, indem wir uns ohne Beschränkung der Allgemeinheit auf Flächen beschränken. Bisher sind wir davon ausgegangen, Punkte zu triangulieren, wie in Abbildung 7.6 b) dargestellt. Die einfachste Möglichkeit, zu einer solchen Triangulation zu gelangen, ist die Generierung von Zentroiden für jede Fläche. Die Triangulierung in Abbildung 7.6 b) ist so aus den Flächen in Abbildung 7.6 a) erzeugt worden. Wir können also auf diese Weise eine Punktmenge erzeugen, die wir direkt mit dem oben beschriebenen Verfahren bearbeiten können.

Eine andere Möglichkeit, die besser die Form und Lage der Flächen berücksichtigt, ist die Berechnung einer sogenannten *Constrained-Delaunay-Triangulation*, d.h. man führt Zwangsbedingungen in Form von Kanten und Flächen ein, die in der Triangulation enthalten sein müssen und von keinen neu berechneten Kanten geschnitten werden dürfen. Eine solche Triangulierung, wie in Abbildung 7.6 c) dargestellt, ist im strengen Sinne keine Delaunay-Triangulation mehr, aber das stellt in vielen Fällen jedoch keine Einschränkung dar. Benötigt man jedoch eine korrekte Delaunay-Triangulation, muss man eine sogenannte *konforme Delaunay-Triangulation* berechnen, die zusätzliche Stützpunkte auf den Linien und Flächenrändern einführt, um das Delaunay-Kriterium zu erfüllen. Aus der so erhaltenen Delaunay-Triangulation können dann die anderen Nachbarschaftsgraphen (GG, RNG, MST, NNG) abgeleitet werden, die dann constrained oder konforme Nachbarschaftsgraphen darstellen. Der Initialisierungsschritt unseres Verfahrens muss dann wie folgt geändert werden:

- Weise allen Knoten die gleiche Cluster-Id zu, die Stützpunkte der gleichen Linie oder Fläche sind.
- Entferne alle Kanten, die Stützpunkte des gleichen Objekts miteinander verbinden.

Nach dieser geänderten Initialisierung kann unser Verfahren unverändert angewendet werden.

[6] Entfernen bedeutet, die Kanten als *disabled* zu markieren und alle Berechnungsschritte des Algorithmus berücksichtigen nur Kanten, die als *enabled* und *aktiv* markiert sind.



(a) Delaunay-Triangulation der Flächenzentroide

(b) Delaunay-Triangulation einer Punktmenge abgeleitet aus a)

(c) Constrained-Delaunay-Triangulation der Flächenpolygone

Abbildung 7.6: Delaunay-Triangulation polygonförmiger Objekte

7.8 Verallgemeinerung auf qualitative Daten

Mit rein qualitativen Daten kann man, wie in Kapitel 5 beschrieben, nicht „rechnen“, d.h. man kann nicht aus gegebenen Werten einen neuen Wert ableiten. Angenommen, wir haben die fünf Objekte *A, B, C, D* und *E* und die dazu gehörige 5×5-Distanzmatrix (Tabelle 7.1) gegeben, dann können wir sofort den NNG (6.2.5) und den MST (6.2.7) bestimmen und auf diesen Graphen unser Clusterverfahren anwenden. Mit Hilfe der Gleichungen 6.5 und 6.7 kann man auch den RNG und den GG bestimmen, da beide Gleichungen eine nicht geometrische Interpretation erlauben, die allein auf den Werten der Distanzmatrix aufbaut. Abbildung 7.7 zeigt die aus Tabelle 7.1 abgeleiteten Nachbarschaftsgraphen.

Wie jedoch bestimmt man für diese fünf Objekte das für die Delaunay-Triangulation benötigte Umkreiskriterium? Im Falle eines zweidimensionalen Vektorraums ist z.B. für drei Punkte a, b und c die Gleichung 7.10 zu prüfen, um festzustellen, ob ein Punkt d innerhalb oder außerhalb des Umkreises von a, b und c liegt (analog zu Gleichung 7.10 lässt sich für jede Dimension n eine entsprechende $(n+1)$-dimensionale Determinante angeben).

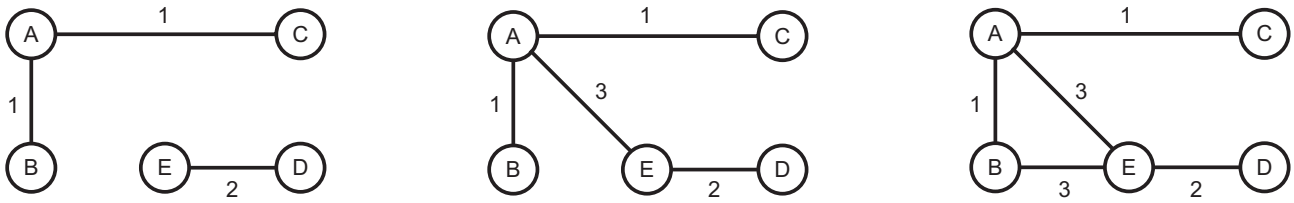
Eine solche Determinante existiert jedoch nicht in unserem qualitativen Fall. Selbst wenn wir die Dimension des Problems kennen würden, so könnten wir trotzdem nicht eine dementsprechende n-dimensionale Sphäre bestimmen. Was wäre denn der Mittelpunkt dieser Sphäre bzgl. *A, B, C, D* und *E*? Im rein qualitativen Fall ist es nicht einmal möglich, einen Mittelwert zu bestimmen. Es ist jedoch im Falle ordinaler Daten möglich, den Median anzugeben. Im Falle von Mengen mit gerader Anzahl von Elementen müsste man sich aber für den oberen oder unteren Median entscheiden, da der Pseudomedian wie der Mittelwert nicht bestimmbar ist. Im qualitativen Falle sind somit nur solche Cluster-Verfahren anwendbar, die keine Cluster-Repräsentanten (*Medoid*) benötigen oder diesen durch „Auswahl“ aus den Clusterelementen bestimmen und ihn nicht „berechnen“.

$$\begin{vmatrix} a_x & a_y & a_x^2+a_y^2 & 1 \\ b_x & b_y & b_x^2+b_y^2 & 1 \\ c_x & c_y & c_x^2+c_y^2 & 1 \\ d_x & d_y & d_x^2+d_y^2 & 1 \end{vmatrix} = \begin{vmatrix} a_x-d_x & a_y-d_y & (a_x^2-d_x^2)+(a_y^2-d_y^2) \\ b_x-d_x & b_y-d_y & (b_x^2-d_x^2)+(b_y^2-d_y^2) \\ c_x-d_x & c_y-d_y & (c_x^2-d_x^2)+(c_y^2-d_y^2) \end{vmatrix} \quad \begin{cases} > 0, & d \text{ innerhalb} \\ = 0, & d \text{ Rand} \\ < 0, & d \text{ außerhalb} \end{cases} \tag{7.10}$$

Unser Clusterverfahren kann also auch auf rein qualitative Daten angewendet werden, wenn man auf die Delaunay-Triangulation verzichtet und für die Berechnung des RNG und GG eine Zeitkomplexität von $O(n^3)$ in Kauf nimmt. Da man bei rein qualitativen Daten nur auf der Distanzmatrix (Ähnlichkeitsmatrix) arbeiten kann, müssen für alle $n^2/2$ Distanzwerte, der gegebenen Distanzmatrix, n Vergleiche durchgeführt werden.

	A	B	C	D	E
A	0	1	1	4	3
B	1	0	2	∞	3
C	1	2	0	5	4
D	4	∞	5	0	2
E	3	3	4	2	0

Tabelle 7.1: Beispiel für eine Distanzmatrix



(a) NNG (b) MST (c) RNG = GG

Abbildung 7.7: Die aus Tabelle 7.1 abgeleiteten Nachbarschaftsgraphen.

7.8.1 Verknüpfung von vektoriellen und qualitativen Daten

Lassen sich Daten in einen *vektoriellen* und in einen *qualitativen* Anteil aufteilen, wie es im allgemeinen bei raumbezogenen Daten der Fall ist, dann bieten sich zwei Möglichkeiten an, um unser Verfahren auf solche Daten anzuwenden.

1. Qualitativ gewichtete Triangulierung der vektoriellen Daten.
2. Erweiterung der n-dimensionalen vektoriellen Daten um eine Dimension, die die Distanz oder Ähnlichkeit der qualitativen Daten zu einem *Referenzobjekt* darstellt.

Im ersten Fall erzeugen wir die Nachbarschaftsgraphen (NNG, MST, RNG, GG und DT) für den n-dimensionalen vektoriellen Datenanteil und gewichten dann die berechneten Kantenlängen (Distanzen) mit der Distanz des qualitativen Anteils. Wie diese Gewichtung durchgeführt wird (multiplikativ oder additiv), hängt vom jeweiligen Anwendungsfall ab. Unser Clusterverfahren kann dann auf den so berechneten gewichteten Graphen angewendet werden. Besteht der vektorielle Datenanteil aus polyederförmigen Objekten, hat man wie in Abschnitt 7.7 vorzugehen.

Im zweiten Fall geht man davon aus, dass man ein sogenanntes Referenzobjekt angeben kann und verwendet die qualitative Distanz der Objekte zu dem gegebenen Referenzobjekt, um den n-dimensionalen Raum des vektoriellen Anteils um eine Dimension zu erhöhen. Jedes Objekt ist dann eindeutig in einem $(n+1)$-dimensionalen Raum beschrieben, auf den dann unser Verfahren angewendet werden kann. Diese Vorgehensweise ist natürlich immer nur dann anwendbar, wenn ein solches Referenzobjekt geeignet anzugeben ist und eine Gruppierung relativ zu diesem Objekt gewünscht ist. Ein Beispiel dafür wäre die Gruppierung von Gebäuden relativ zu Wohngebäuden, um Gebäudegruppen zu erhalten, die mehr oder weniger ähnlich zu Wohngebäuden sind, etwa um die Siedlungsstruktur einer Stadt oder Region zu analysieren.

7.9 Berechnung der Randbeschreibung eines Clusters

Die Randbeschreibung eines Clusters lässt sich anhand der inneren und äußeren Kantenmenge sehr einfach ermitteln, wenn eine Delaunay-Triangulation zur Verfügung steht. Jede innere Kante eines Clusters, die zu einem Dreieck der Delaunay-Triangulation gehört, dessen beide anderen Kanten äußere Kanten des Clusters sind, ist eine Kante der Randbeschreibung des Clusters. In einem weiteren Schritt muss diese Kantenmenge

daraufhin analysiert werden, ob die Kanten offene oder geschlossene Polygonzüge oder Dreiecke bilden. Im Falle eines offenen Polygonzugs stellt der Cluster eine „linienförmige“ Struktur dar. Handelt es sich dagegen um einen geschlossenen Polygonzug, so ist der Cluster „flächenförmig“. Bilden die Kanten dagegen Dreiecke, so handelt es sich um einen „volumenförmigen“ Cluster.

Kapitel 8

Evaluierung des HPGCL-Algorithmus

Wie kann man nun die Korrektheit und die Performance eines Clusterverfahrens überprüfen? Die einfache Antwort lautet durch Testdatensätze, was sich wie folgt begründen lässt: erstens ist der beschriebene Algorithmus eine Heuristik, die unser Modell der Cluster mit homogener Dichte möglichst gut approximieren soll und eine Heuristik lässt sich nun einmal nicht beweisen, denn zu einem formalen Beweis müsste auch eine formale Theorie existieren. Zweitens kann ein Modell der Realität immer nur anhand realer Daten durch den Menschen verifiziert (bestätigt) werden. Leider ist diese Verifizierung auch nur subjektiv, denn die Gruppierung von „ähnlichen“ Objekten ist ein kognitiver Prozess, der nicht formal eindeutig definiert, sondern nur so gut wie möglich beschrieben werden kann. Zu einer solchen Beschreibung gehören immer mehrere Kriterien. Wir haben in unserem Fall als Kriterien die Nachbarschaft (bzgl. der Nachbarschaftsgraphen) und die Homogenität (der Abstände) der Objekte eines Clusters gewählt.

8.1 Testdaten

Dass die Einteilung von Punktmengen in homogene Gruppen selbst für den Menschen nicht eindeutig und einfach ist, wollten wir anhand eines Tests überpüfen. Für diesen Test verwendeten wir die in Abbildung 8.1a und 8.1b dargestellten Punktmengen und baten 10 Mitarbeiter des Instituts für Photogrammetrie der Universität Stuttgart (ifp), diese Punktmengen intuitiv in zusammengehörige Gruppen einzuteilen. Die Ergebnisse dieser 10 manuellen Auswertungen sind in der Tabelle 8.1 und im Anhang A dargestellt.

Wie man aus Tabelle 8.1 und den Abbildungen aus Anhang A sofort erkennt, sind für jeden Datensatz jeweils 10 verschiedene Ergebnisse[1] erzielt worden. Einige Datensätze unterscheiden sich nur wenig, andere dagegen erheblich. Die manuellen Auswertungen liefern uns zwar keine eindeutige Referenzzerlegung, sie bieten jedoch trotzdem eine sehr gute Vergleichsmöglichkeit. Dabei fällt auf, dass bei den manuellen Auswertungen nicht nur die homogene Anordnung der Punkte eine Rolle spielte, sondern auch die Form der Cluster. Rechteckige Formen wurden bevorzugt. Im ersten Datensatz wurden im Mittel zwischen 11 und 15 Cluster erkannt und im zweiten Datensatz zwischen 11 und 22 Cluster. Die große Varianz im zweiten Datensatz ist umso erstaunlicher, da er auf den ersten Blick erheblich flächenförmiger wirkt als der erste Datensatz, trotzdem schwankten im zweiten Datensatz die Auswertungen zwischen stark generalisierend (Abb. A.6a) und detaillierter Zerlegung (Abb. A.6b).

[1] Für alle folgenden Clusterdarstellungen gilt, dass benachbarte Punkte mit gleicher Farbe und gleicher Größe zu einem Cluster gehören. Schwarze Punkte bei den manuellen Auswertungen gelten als keinem Cluster zugeordnet. Wenn möglich, wurden auch die inneren Kanten eines Clusters mit der gleichen Farbe dargestellt.

(a) Testdatensatz 1 (b) Testdatensatz 2

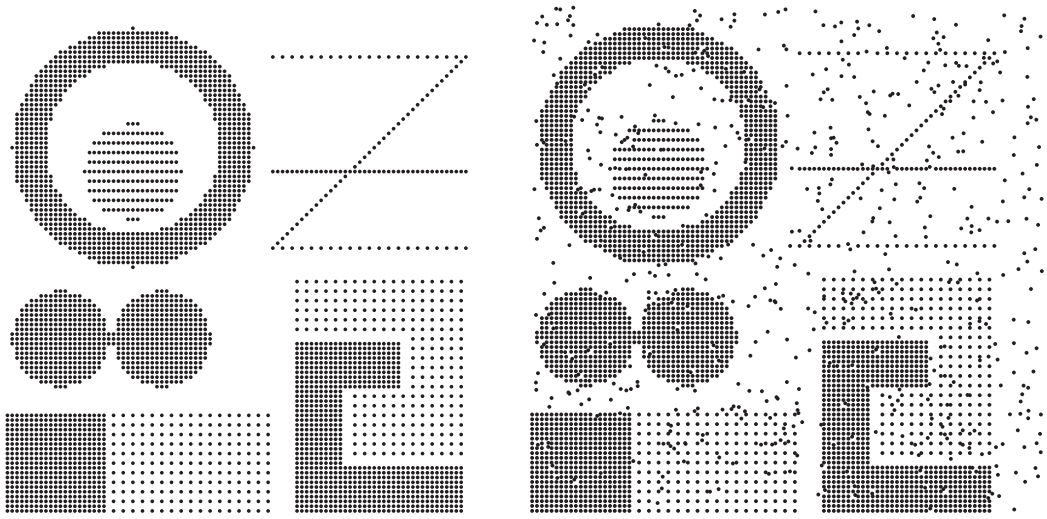
Abbildung 8.1: Testdaten zur manuellen Auswertung

Tabelle 8.1: Ergebnisse der manuellen Auswertung zweier Testdatensätze

	Manueller Test 1		**Manueller Test 2**	
Auswertung	#Cluster	#Unklassifizierter Objekte	#Cluster	#Unklassifizierter Objekte
1	11	0	16	9
2	17	4	14	9
3	6	0	6	0
4	13	2	29	13
5	13	2	25	4
6	14	3	17	15
7	13	4	10	10
8	23	1	23	29
9	11	0	12	14
10	16	3	21	29
Min	6	0	6	0
Max	23	4	29	29
Median	**13**	**2**	**16,5**	**11,5**
MAD	**2**	**1,5**	**5,5**	**3**
Mittelwert	13,7	1,9	17,3	13,2
Sigma	4,22	1,513	6,812	8,987

Um eine eindeutige Referenzzerlegung zu erhalten, haben wir drei künstliche Datensätze erzeugt, die solche Clusterformen enthalten, die wir durch den HPGCL-Algorithmus erkennen wollen. In Abbildung 8.2a ist der erste Testdatensatz abgebildet. Dieser Datensatz enthält einen linienförmigen Cluster in Form eines Z, nah benachbarte Linien-Cluster innerhalb eines Ring-Clusters, konkave und verschränkte Cluster unterschiedlicher Dichte, sich berührende Cluster gleicher Dichte sowie benachbarte Cluster unterschiedlicher Dichte.

Als zweiten Testdatensatz haben wir den ersten mit Rauschen überlagert (Abb. 8.2b), d.h. es wurden zufällig Punkte hinzugefügt, entfernt und verschoben. Dieser Datensatz dient dazu, eine Abschätzung der Rauschempfindlichkeit zu ermöglichen. Das besondere Interesse liegt dabei darin, ob die gleichen Regionen wie im ersten Testdatensatz als Cluster erkannt wurden. Eine Unterscheidung der Punkte in Rauschen und Cluster ist nicht unser Ziel. Es kann auch nicht unser Ziel sein, da unser Clustermodell Rauschen nicht modelliert. Unser Modell interpretiert homogene Regionen als Cluster und diese homogenen Regionen können natürlich auch aus Rauschen bestehen.



(a) ohne Rauschen (b) mit Rauschen

Abbildung 8.2: Künstlich erzeugte Punktmenge

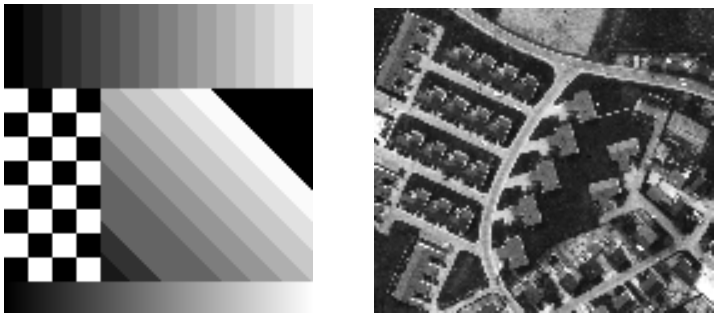
Abbildung 8.3: Gebäude-Zentroide eines Gebiets von Stuttgart-Vaihingen

Als dritten künstlichen Datensatz haben wir das in Abbildung 8.4a dargestellte Grauwertbild erzeugt. Es soll einerseits die Anwendung unseres Verfahrens auf Multispektraldaten verdeutlichen und andererseits eine Einschätzung der Differenziertheit unseres Verfahrens liefern. Dazu haben wir grobe und feine Grauwertverläufe sowie ein Schachbrettmuster erzeugt. Abbildung 8.17a auf Seite 100 zeigt den dreidimensionalen Objektraum des Grauwertbildes.

Neben den künstlichen Testdaten haben wir natürlich auch reale Daten verwendet. Der Laserscan eines Tiefziehblechs (Abb. 8.5) diente uns als reales Beispiel für einen dreidimensionalen Datensatz. Eine Schwarzweiß-Luftbildaufnahme (Abb. 8.4b) diente als Beispiel für reale Multispektraldaten und als weiterer Testdatensatz stand uns ein Ausschnitt des Stuttgarter Gebäudeinformationssystems vom Gebiet Stuttgart-Vaihingen zur Verfügung[2]. Aus diesem Datensatz extrahierten wir die Gebäudegrundrisse, berechneten für jedes dieser Polygone den Schwerpunkt und verwendeten die Menge dieser Schwerpunkte (siehe Abbildung 8.3), sowie auch Ausschnitte daraus, als Beispiel für geographisch verteilte Objekte. Die Testdaten für die manuelle Auswertung sind ebenfalls Ausschnitte aus diesem Datensatz.

Zur Untersuchung unseres Verfahrens haben wir acht verschiedene Modi (Tabelle 8.2) definiert und für jeden Modus eine Berechnung durchgeführt. Die in Tabelle 8.2 aufgezählten Modi (Spalte 1) unterscheiden sich jeweils

[2]Freundlicherweise vom Stuttgarter Stadtmessungsamt zur Verfügung gestellt.



(a) Testbild (b) Luftbild

Abbildung 8.4: Verwendete Bilder

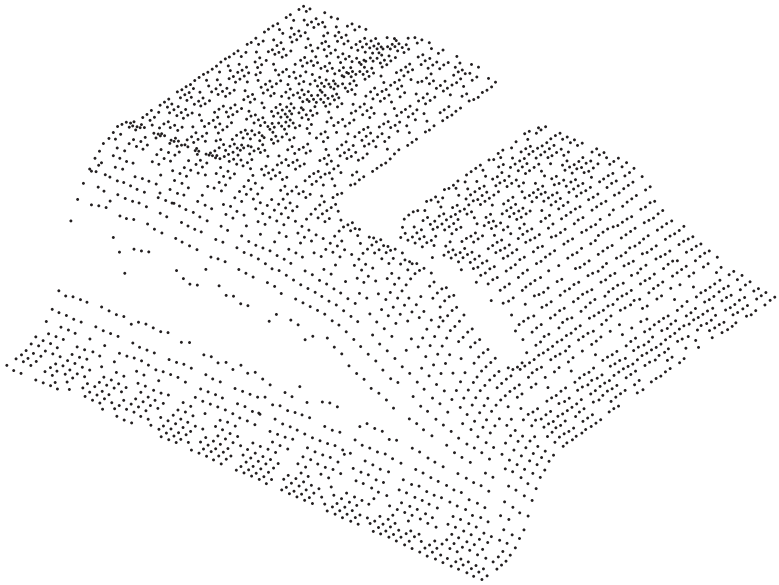


Abbildung 8.5: 3D-Abstandsdaten eines Tiefziehblechs

Modus	Iteration	Einzelschritt	Maximiere Clusterabstände
1	nein	nein	nein
2	ja	nein	nein
3	nein	nein	ja
4	ja	nein	ja
5	nein	ja	nein
6	ja	ja	nein
7	nein	ja	ja
8	ja	ja	ja

Tabelle 8.2: Verwendete Testmodi

in den drei verwendeten Varianten des HPGCL-Algorithmus. *Iteration* steht für die Verwendung der iterativen Erweiterung unseres Algorithmus[3], d.h. der HPGCL-Algorithmus wird nur einmal durchlaufen oder solange wiederholt bis die Grenzzerlegung erreicht wurde. *Einzelschritt*[4] sagt aus, ob ein Cluster nur mit dem nächsten kompatiblen Cluster vereinigt wird oder mit allen kompatiblen Clustern. *Maximiere Clusterabstände* steht für die optionale Maximierung des äußeren Clusterabstands[5]. In Anhang B sind tabellarisch alle Messergebnisse aufgelistet.

8.2 Auswirkung der Nachbarschaftsgraphen auf die Anzahl der Cluster

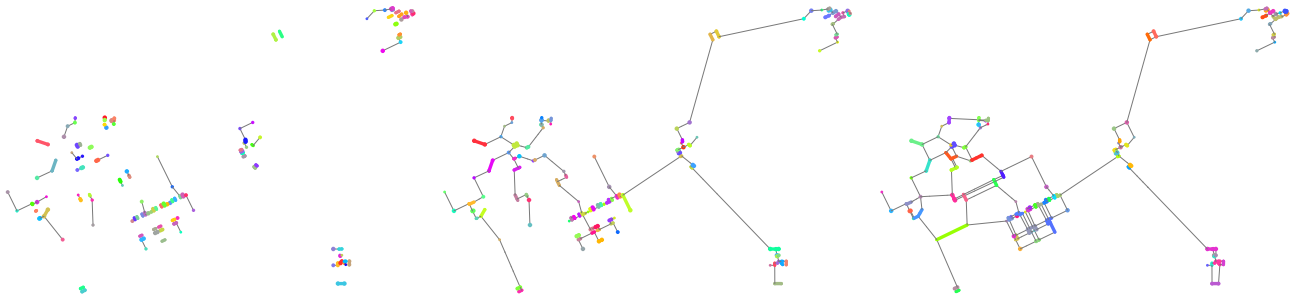
Dass die Verwendung der einzelnen Nachbarschaftsgraphen und deren Hierarchie kein Selbstzweck ist, sondern in der Tat einen wesentlichen Einfluss auf den Gruppierungsprozess hat, zeigen die folgenden Ergebnisse. Dazu haben wir zuerst unser Verfahren auf den gleichen Datensatz jeweils nur mit einem der Nachbarschaftsgraphen angewendet[6] und danach haben wir den HPGCL-Algorithmus auf den selben Datensatz mit den Teilhierarchien NNG, $\mathsf{NNG} - \mathsf{MST}$, $\mathsf{NNG} - \mathsf{MST} - \mathsf{RNG}$, $\mathsf{NNG} - \mathsf{MST} - \mathsf{RNG} - \mathsf{GG}$ und $\mathsf{NNG} - \mathsf{MST} - \mathsf{RNG} - \mathsf{GG} - \mathsf{DT}$ angewendet.

In Abbildung 8.6 sind die Ergebnisse der einzelnen Graphen für einen Beispieldatensatz dargestellt. Wie man sieht, ist die Anzahl der gefundenen Cluster umso kleiner, je größer der Nachbarschaftsgraph ist. Dies entspricht genau dem erwünschten parameterfreien *Lupeneffekt* (Buffergrowing), den unser Clusterverfahren besitzen soll, d.h. in der untersten Hierarchiestufe (NNG) erhält man eine detaillierte Gruppierung der Ausgangsmenge und in der obersten Hierarchiestufe (DT) eine gröbere Aufteilung.

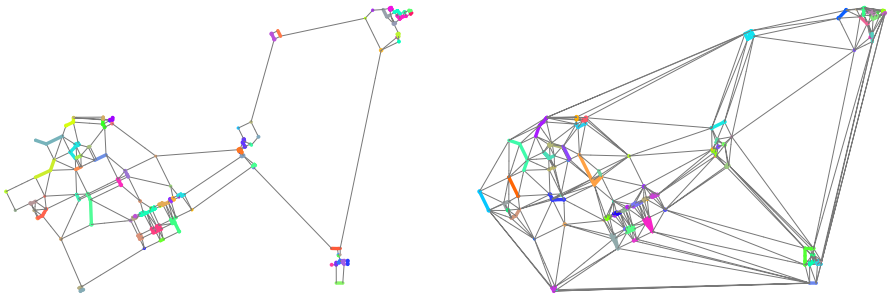
Die Ergebnisse bei der Anwendung der einzelnen Teilhierarchien auf den gleichen Datensatz sind in Abbildung 8.7 dargestellt. Wie man sieht, nahm die Anzahl der Cluster in der angegebenen Reihenfolge ebenfalls ab. Gleichzeitig verringerte sich die Anzahl der Cluster noch einmal gegenüber den einzelnen Graphen.

Diese beiden Ergebnisse beziehen sich auf den Modus 1, d.h. es wurde keine Iteration durchgeführt. Das Verhältnis von Clusteranzahl und verwendeten Graphen oder Teilhierarchien, bei nicht-iterativer Anwendung des HPGCL-Algorithmus, wird in den Abbildungen 8.10a und 8.11b noch deutlicher. Für diese beiden Diagramme wurden verschieden große Datensätze in den Modi 1, 3, 5 und 7 von unserem Algorithmus bearbeitet. Im Falle der Anwendung einzelner Graphen zeigt sich eine klare Unterscheidung zwischen der expliziten Maximierung des äußeren Clusterabstandes (Modus 3 und 7) und dem Verzicht auf diese Bedingung (Modus 1 und 5). In den Modi 1 und 5 nimmt die Clusteranzahl annähernd linear ab, dagegen nimmt sie im Modus 3 und 7 vom NNG zum MST ab und steigt dann wieder an. Bei der Verwendung der Teilhierarchien zeigt sich diese Unterscheidung nicht mehr. In allen Fällen nimmt die Anzahl der Cluster näherungsweise linear ab, aber auch hier liefern die Modi 3 und 7 eine größere Anzahl von Clustern als die Modi 1 und 5, was zu erwarten war.

[3] Siehe Abschnitt 7.5.2.
[4] Siehe Abschnitt 7.5.1 Fußnote zu Schritt 5.
[5] Siehe Abschnitt 7.5.1 Schritt 4: Optionaler Test nach Definition 7.4.5.
[6] Der HPGCL-Algorithmus kann natürlich auch nur mit einem einzelnen Graphen oder einer beliebigen Auswahl von Graphen arbeiten.

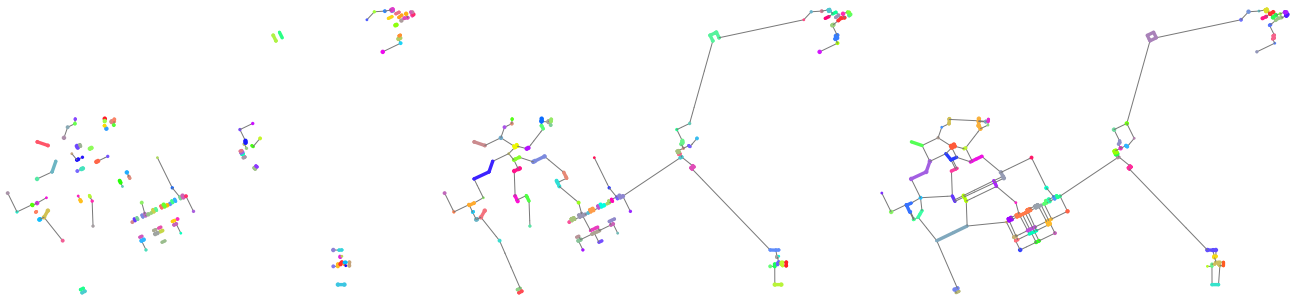


(a) NNG: 126 Cluster (b) MST: 116 Cluster (c) RNG: 101 Cluster

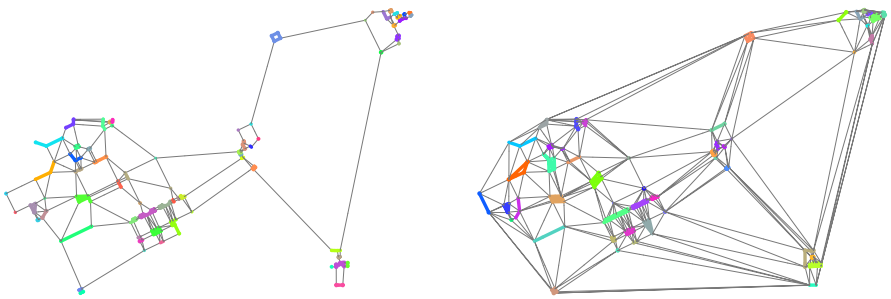


(d) GG: 85 Cluster (e) DT: 74 Cluster

Abbildung 8.6: Auswirkung der einzelnen Graphen auf die Clusteranzahl (Modus 1)

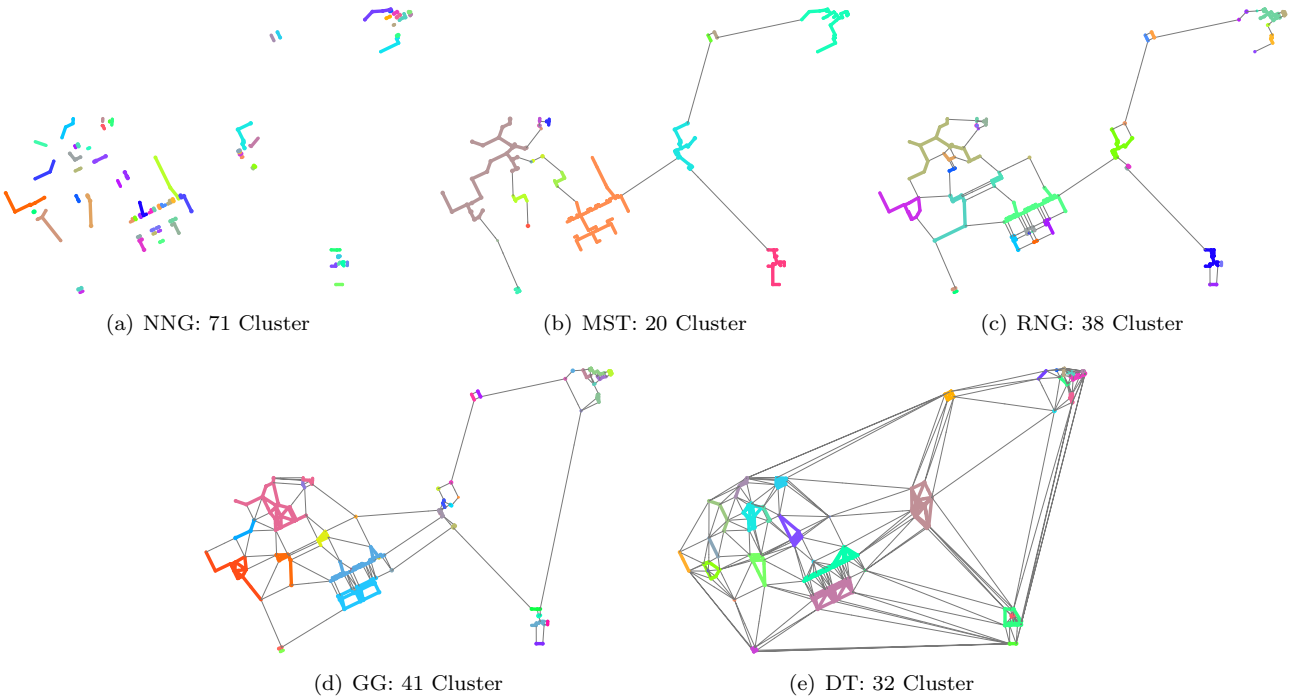


(a) NNG: 126 Cluster (b) NNG-MST: 92 Cluster (c) NNG-RNG: 80 Cluster



(d) NNG-GG: 70 Cluster (e) NNG-DT: 50 Cluster

Abbildung 8.7: Auswirkung der Graphen-Hierarchie auf die Clusteranzahl (Modus 1)



(a) NNG: 71 Cluster (b) MST: 20 Cluster (c) RNG: 38 Cluster

(d) GG: 41 Cluster (e) DT: 32 Cluster

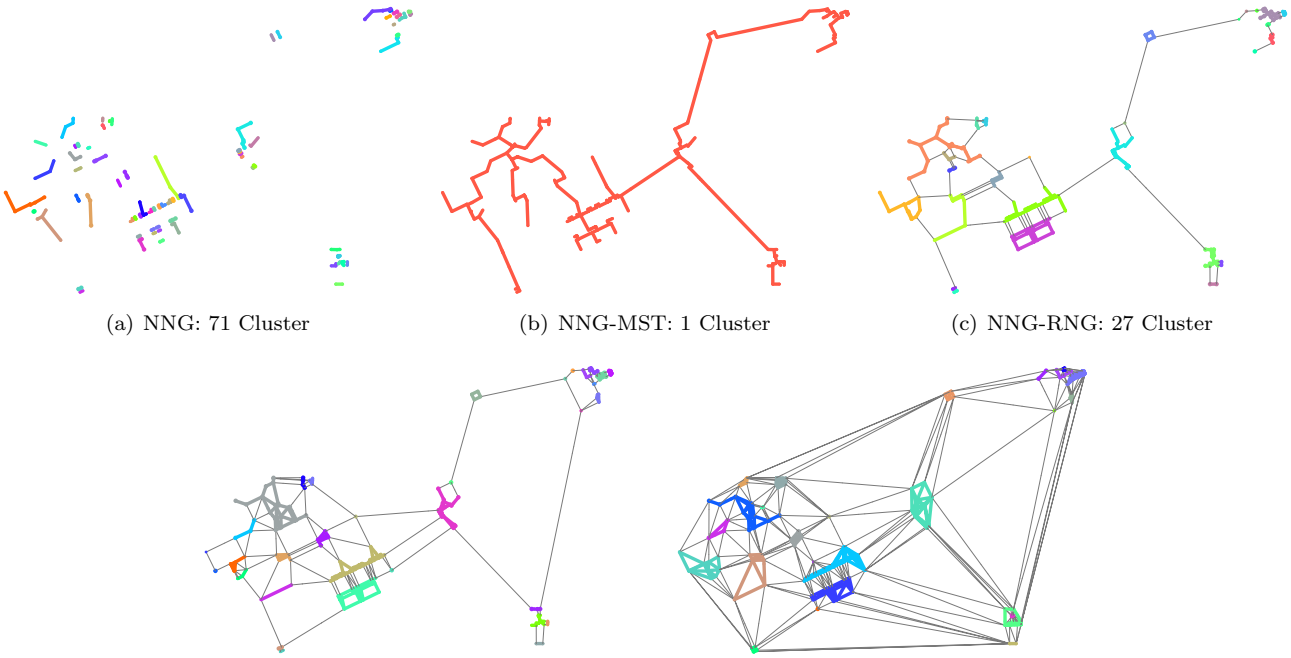
Abbildung 8.8: Der Lupeneffekt gilt nicht im Falle des iterativen Clustering (Modus 2–einzelner Graph)

Ein gänzlich anderes Verhalten zeigte sich im Falle der iterativen Wiederholung des HPGCL-Algorithmus. Weder bei der Anwendung der einzelnen Graphen (Abb. 8.8), noch bei den Teilhierarchien (Abb. 8.9) zeigte sich der Lupeneffekt. Die Anwendung der Modi 2, 4, 6 und 8 auf unterschiedlich große Datensätze zeigte den in den Abbildungen 8.10b und 8.11b dargestellten interessanten Zusammenhang auf. In beiden Fällen zeigte sich eine eindeutig größere Anzahl von Clustern bei der Maximierung des äußeren Clusterabstands, der besonders deutlich im Falle der Teilhierarchien (Abb. 8.11b) hervor tritt. Weiterhin zeigte sich, dass der MST und die Teilhierarchie NNG – MST besonders stark gruppierend wirken. Die Teilhierarchie NNG – MST kann häufig zu einem einzigen Cluster führen. Wie bei der nicht-iterativen Anwendung des HPGCL-ALgorithmus zeigt sich bei der iterativen Anwendung ein Unterschied bei der Maximierung des äußeren Abstands. Im Falle der einzelnen Graphen steigt die Anzahl der Cluster vom MST zur DT kontinuierlich an. Dagegen steigt sie im Falle der Teilhierarchien bis zum GG wieder an, um dann zur DT wieder abzufallen – was dem Verhalten der Modi 2 und 4 bei Anwendung der einzelnen Graphen und der Teilhierarchien entspricht.

Es existiert ein weiteres interessantes hierarchisches Verhältnis zwischen der Anzahl der gefundenen Cluster und der Anwendung eines einzelnen Graphen mit und ohne Iteration und der Anwendung einer Teilhierarchie mit und ohne Iteration. Die Abbildung 8.12 zeigt dieses Verhältnis für fünf verschiedene Datensätze. Für einen Graphen X ergibt sich somit eine $\geq -$*Relation* bzgl. der Anzahl der Cluster in der folgenden Reihenfolge:

1. Einzelner Graph X ohne Iteration.
2. Teilhierarchie NNG – X ohne Iteration.
3. Einzelner Graph X mit Iteration.
4. Teilhierarchie NNG – X mit Iteration.

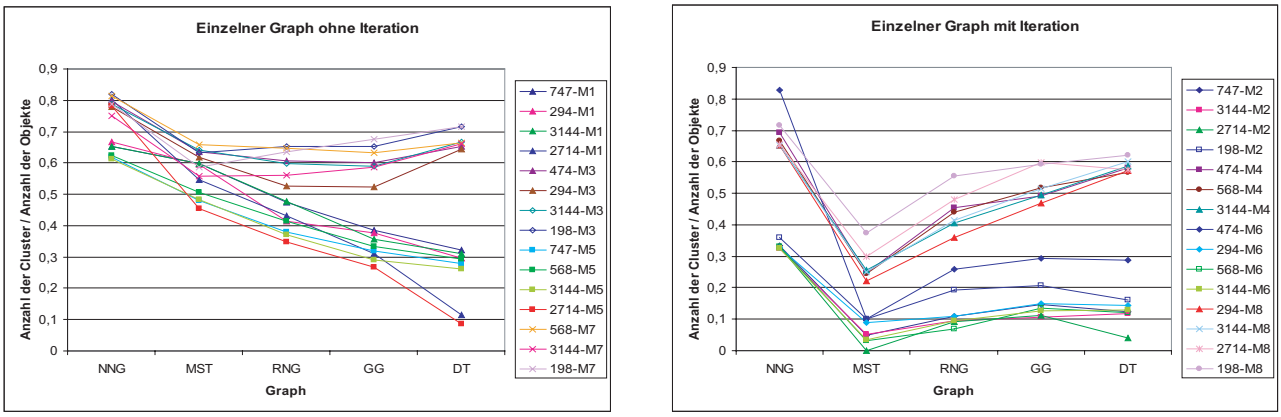
Die in den Abbildungen 8.10, 8.11 und 8.12 dargestellten Kurven können zur Entscheidung verwendet werden, welchen Graphen, welche Teilhierarchie oder welchen Modus man verwenden soll, um eine bestimmte Mindestreduktion der Ausgangsmenge zu erreichen. Die Kurven zeigen z.B., dass die Anwendung der Hierarchie NNG – DT ohne Iteration im Mittel eine Reduktion um 70 Prozent bewirkt. Bei der iterativen Anwendung reduziert sich dagegen die Anzahl der Objekte zwischen 70 und 90 Prozent, je nachdem, ob man den äußeren Clusterabstand maximiert oder nicht.



(a) NNG: 71 Cluster (b) NNG-MST: 1 Cluster (c) NNG-RNG: 27 Cluster

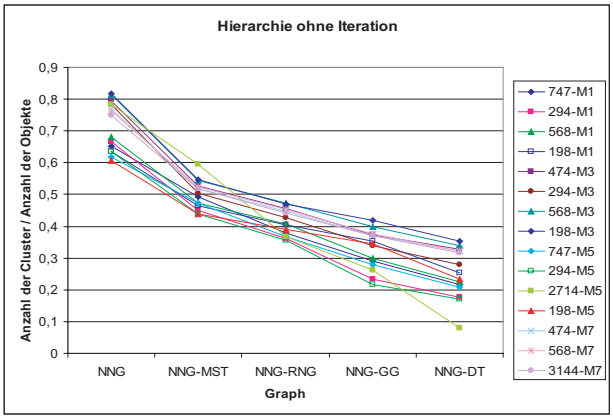
(d) NNG-GG: 33 Cluster (e) NNG-DT: 25 Cluster

Abbildung 8.9: Der Lupeneffekt gilt nicht im Falle des iterativen Clustering (Modus 2–Graphen-Hierarchie)

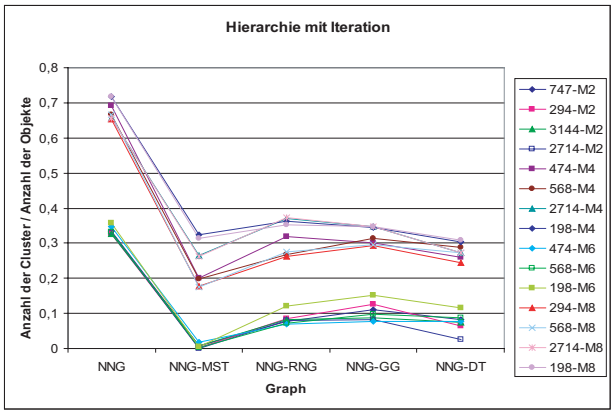


(a) ohne Iteration (b) mit Iteration

Abbildung 8.10: Verhältnis von Clusteranzahl zu Objektanzahl in Abhängigkeit vom verwendeten Graphen

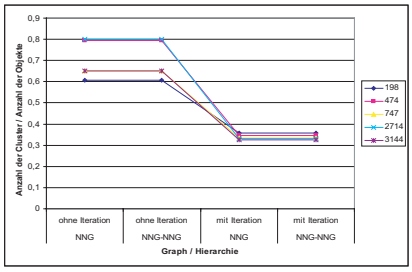


(a) ohne Iteration

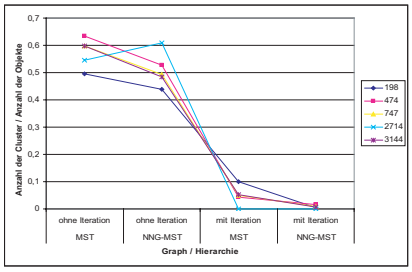


(b) mit Iteration

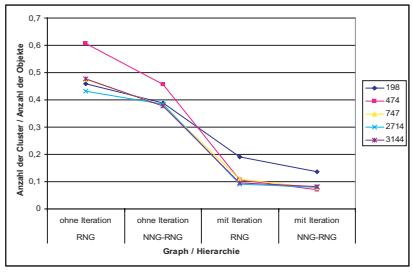
Abbildung 8.11: Verhältnis von Clusteranzahl zu Objektanzahl in Abhängigkeit von der verwendeten Hierarchie



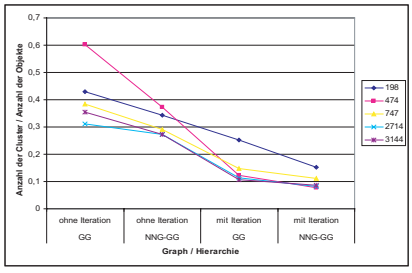
(a) NNG



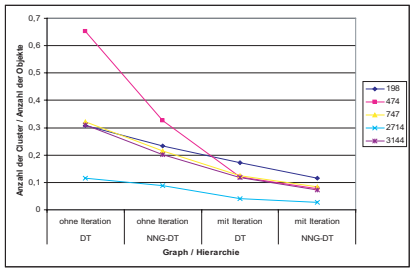
(b) MST



(c) RNG



(d) GG



(e) DT

Abbildung 8.12: Verhältnis von Clusteranzahl zu Objektanzahl in Abhängigkeit des einzelnen Graphen und der zugehörigen Teilhierarchie

8.3 Ergebnisse für die künstlichen Testdaten

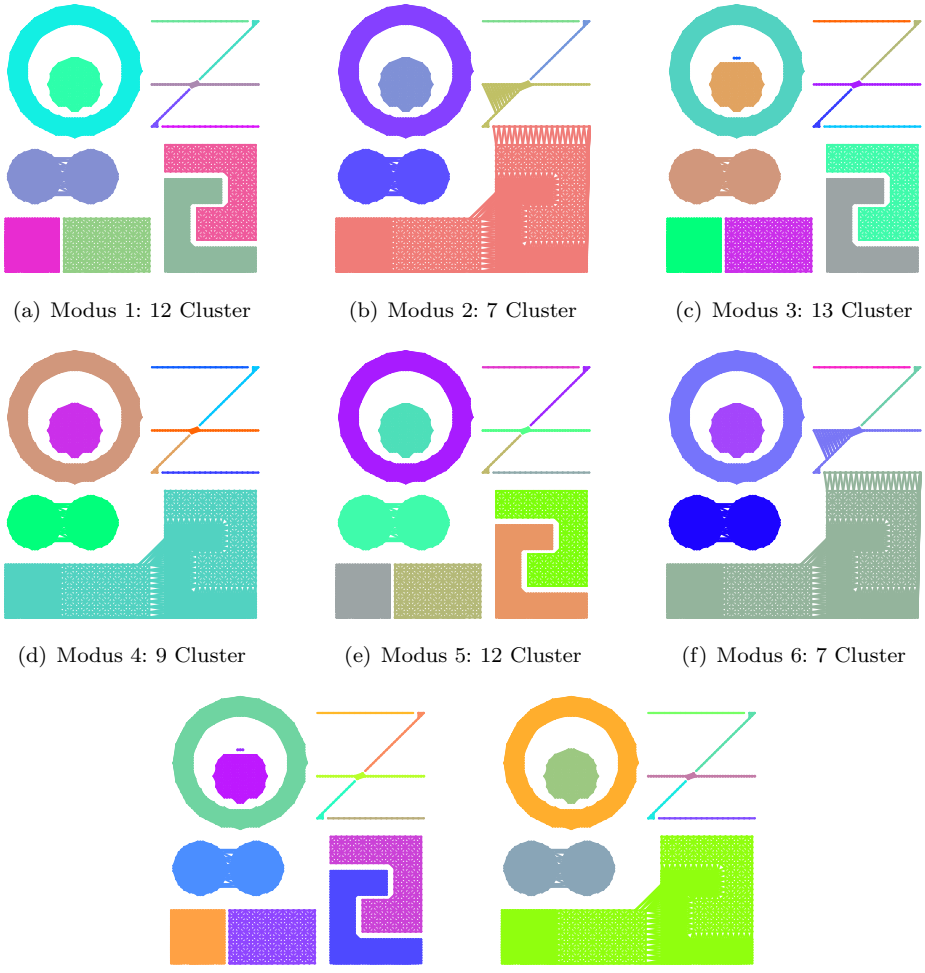
8.3.1 Punktmuster

Im künstlichen Testdatensatz sollte, nach unserer Meinung, ein optimales Cluster-Verfahren neun Cluster erkennen können. Den Kreisring, die Kreisfläche innerhalb des Kreisringes, das „Z“, jede der beiden sich berührenden Kreisflächen unterhalb des Kreisringes, die quadratische und rechteckige Fläche sowie die beiden ineinander verzahnten Flächen. Diese Referenzzerlegung ist jedoch genau betrachtet nicht eindeutig, besonders unter dem Aspekt von Punktmengen mit homogenem Abstand als Cluster. Das „Z“ besteht genau betrachtet aus vier Linien von denen die oberste und unterste den gleichen Punktabstand besitzen – die Assoziation zu einem „Z“ trifft ein menschlicher Operateur aufgrund seines Hintergrundwissens, das unser Verfahren jedoch nicht hat. Die sich berührenden Kreisflächen muss man nicht zwangsweise als getrennte Cluster auffassen. Rein unter dem Aspekt des homogenen Punktabstandes und da auch nicht konvexe Formen erkannt werden sollen, sind die beiden Kreisflächen ein einziger Cluster. Genauso gut kann man die Kreisfläche innerhalb des Kreisringes in elf einzelne horizontale Linien zerlegen, da der vertikale Punktabstand größer als der horizontale ist.

Die Ergebnisse unseres Verfahrens zu diesem Datensatz sind in Abbildung 8.13 dargestellt. Das Ergebnis für den Modus 1 (Abb. 8.13a) kommt unserer Vorstellung von einer Referenzzerlegung am nächsten. Es erkennt den Kreisring als einen Cluster und innerhalb dieses Clusters einen separaten Cluster. Wie man sieht, trennt unser Verfahren sich berührende Cluster gleicher Dichte nicht, was aufgrund unseres Modells auch zu erwarten war. Das „Z“ wird in fünf Segmente zerlegt und nicht als ein einziges Objekt erfasst. Dieses Ergebnis entspricht vollständig unserem Modell, da ja das „Z“ aus vier Linien unterschiedlicher Dichte besteht. Dass die diagonale Linie in zwei Segmente aufgeteilt wurde, ist nach unserem Modell auch korrekt, da unser Modell nur disjunkte Cluster erkennen kann (jeder Punkt kann nur einem Cluster zugeordnet sein) und im Falle von zwei nicht disjunkten Clustern immer mindestens einen von beiden Clustern aufteilt. Interessant ist, dass auch beim iterativen Clustering das „Z“ nie zu einem einzigen Cluster zusammengefasst wird. Das lässt auf eine eindeutige dominante Verteilung dieser Cluster schließen. Die Kreisfläche innerhalb des Kreisringes, die aus mehreren horizontalen Linien besteht, wird im Gegensatz zum „Z“ zu einem Cluster zusammengefasst. Dieses Ergebnis stimmt gut mit der menschlichen Wahrnehmung überein. Dieses Ergebnis lässt sich auch mit unserem Modell begründen, denn im Gegensatz zum „Z“ besitzen alle horizontalen Linien die gleiche Dichte und der Abstand zwischen den Linien ist ebenfalls konstant. Wie man an Abbildung 8.15a sieht, werden alle horizontalen Linien als separate Cluster erfasst, wenn man nur den NNG in unserem Verfahren verwendet. Die Abbildungen 8.15a und b zeigen auch auf, dass im Falle regelmäßig angeordneter Objekte der NNG die wesentliche Cluster-Information enthält, da die meisten Cluster, selbst im verrauschten Datensatz schon auf dieser Hierarchiestufe erkannt wurden. Wie wir noch zeigen werden, ist dies jedoch im Falle allgemein verteilter Objekte nicht der Fall.

Wie man an der Anzahl der Cluster erkennt, bewirkt die Vereinigung eines Clusters immer nur mit seinem nächsten kompatiblen Cluster (Modus 5, 6, 7 und 8) keinen Unterschied zu der gleichzeitigen Vereinigung aller kompatiblen Cluster. Die Ergebnisse in den Modi 2, 4, 6 und 8 zeigen, dass die Grenzzerlegung unseres iterativen Clusterings, gegenüber Standardverfahren, nie ein einziger Cluster aller Punkte ist. Diese Ergebnisse zeigen jedoch auch, dass im Falle von gleichmäßig verteilten Clustern unser Verfahren ebenfalls zu einem einzigen *Megacluster* führen wird, denn in den Modi 2 und 6 sind die unteren vier Cluster und die untere Linie des „Z“ zu einem großen Cluster zusammengefasst worden. Im Modus 4 und 8 wurde dagegen, aufgrund der Maximierung des äußeren Abstands, die untere Linie des „Z“ als eigener Cluster klassifiziert. Die Maximierung des äußeren Abstands erwirkt somit – wie erwartet – eine strengere Zerlegung. Jedoch führt dies, gegenüber der menschlichen Wahrnehmung, wie an den Ergebnissen der Modi 3 und 7 zu sehen, zu unerwarteten Ausreißern, wie der einzelne horizontale Linien-Cluster innerhalb des Kreisringes.

Die Ergebnisse der verrauschten Testdaten (Abb. 8.14) zeigen die Robustheit unseres Verfahrens, da im wesentlichen die Cluster der nicht verrauschten Daten wiedererkannt wurden. Die verrauschten Daten bestätigen somit das Modell der benachbarten Objekte mit homogenem Abstand. Einzelne Objekte, die durch das Rauschen unregelmäßige Abstände, im Gegensatz zu ihren Nachbarn, besitzen, werden nicht zu dem umgebenden Cluster hinzugefügt. Es wäre somit möglich, singuläre oder relativ kleine Cluster, die innerhalb großer Cluster liegen, als Rauschen zu modellieren. Wie man an Abbildung 8.15b sieht, ist bei regelmäßigen Clustern schon allein der NNG sehr robust gegenüber Rauschen. Die Ergebnisse zeigen jedoch auch, dass im Falle einer lokal gehäuften Störung die Ergebnisse natürlich verfälscht werden (siehe z.B. Abb. 8.14a und e), da in diesen Fällen eigenständige homogene Cluster entstehen, die vom Modell her nicht als Fehler interpretiert werden können. In diesen Fällen verbesserte das iterative Clustering, im Sinne der beschriebenen Referenzzerlegung, das Ergebnis,

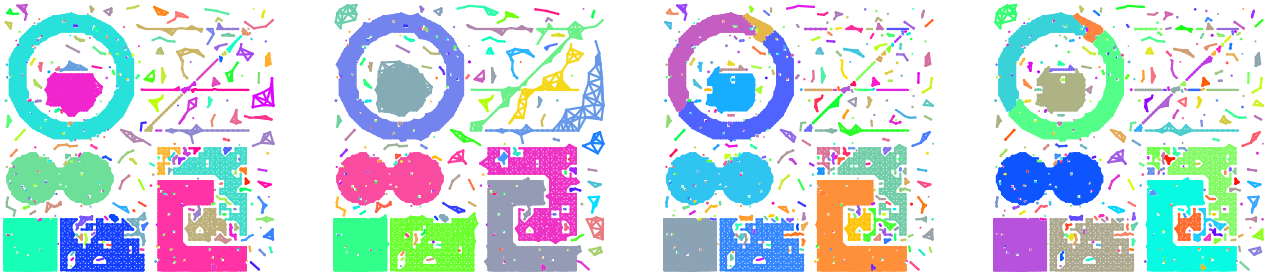


(a) Modus 1: 12 Cluster (b) Modus 2: 7 Cluster (c) Modus 3: 13 Cluster

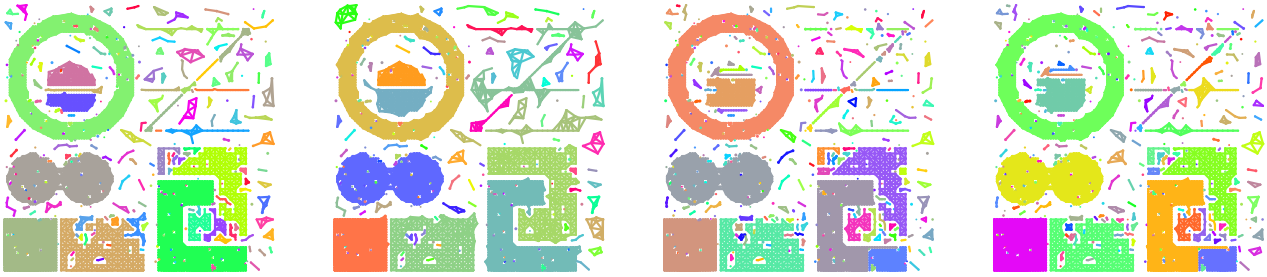
(d) Modus 4: 9 Cluster (e) Modus 5: 12 Cluster (f) Modus 6: 7 Cluster

(g) Modus 7: 13 Cluster (h) Modus 8: 9 Cluster

Abbildung 8.13: Ergebnisse für die künstlichen Testdaten



(a) Modus 1: 280 Cluster (b) Modus 2: 187 Cluster (c) Modus 3: 358 Cluster (d) Modus 4: 328 Cluster



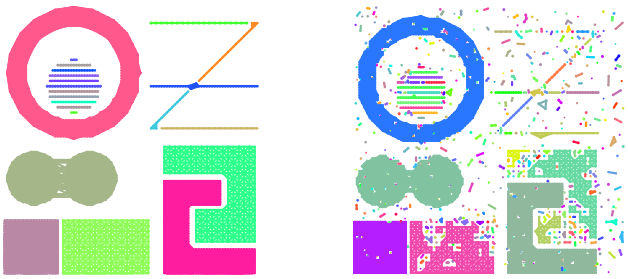
(e) Modus 5: 278 Cluster (f) Modus 6: 195 Cluster (g) Modus 7: 358 Cluster (h) Modus 8: 323 Cluster

Abbildung 8.14: Ergebnisse für die verrauschten künstlichen Testdaten

wie in Abbildung 8.14b und f zu sehen ist. Diese Abbildungen zeigen jedoch auch, dass die Linien-Cluster des „Z“ beim iterativen Clustering erheblich gestört werden. Die linienförmig angeordneten Cluster „laufen aus“, d.h. es entstehen große Cluster durch Vereinigung mit den umgebenden Rauschobjekten. Wie zu erwarten war, sind die linienförmigen Strukturen störanfälliger als die flächenförmigen. Die strenge Bedingung der Vergrößerung der äußeren Abstände führt auch hier zu einer erheblichen Übersegmentierung und zu einer unerwarteten Aufteilung des Kreisringes (Abb. 8.14c und d). Die ausschließliche Vereinigung mit dem nächsten kompatiblen Cluster (Modus 5, 6, 7 und 8) wirkt sich in diesem Fall jedoch aus, leider jedoch nicht einheitlich. Einerseits wird der Kreisring wieder als ein einziger Cluster erkannt, jedoch wird die kreisförmige Fläche stärker gestört als in den Modi 1, 2, 3 und 4.

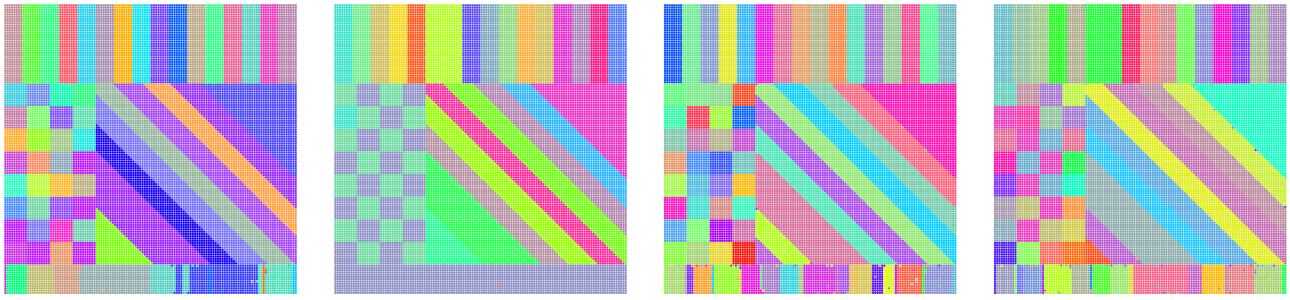
8.3.2 Testbild

Auch bei dem künstlichen Testbild zeigte es sich, dass die strenge Bedingung zur Maximierung des äußeren Abstands eine Verschlechterung des Ergebnisses darstellt, da man vom menschlichen Gesichtspunkt her eher eine

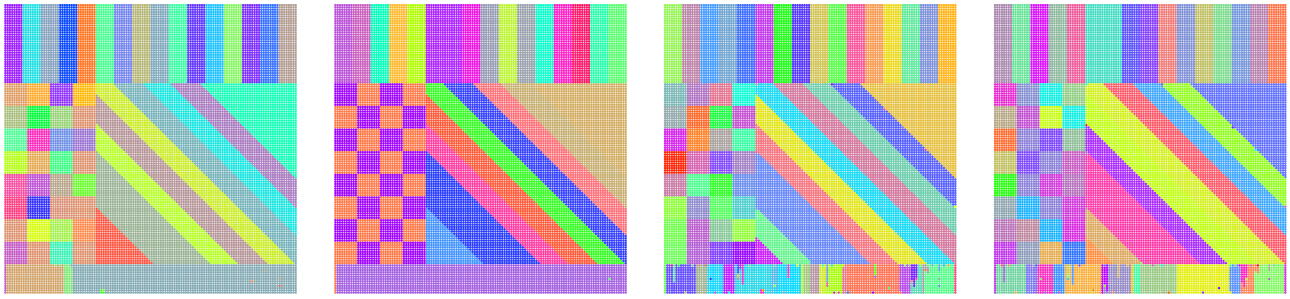


(a) ohne Rauschen (b) mit Rauschen

Abbildung 8.15: NNG-Clustering im Modus 1



(a) Modus 1: 70 Cluster (b) Modus 2: 28 Cluster (c) Modus 3: 138 Cluster (d) Modus 4: 110 Cluster



(e) Modus 5: 61 Cluster (f) Modus 6: 28 Cluster (g) Modus 7: 137 Cluster (h) Modus 8: 119 Cluster

Abbildung 8.16: Ergebnisse für das Testbild

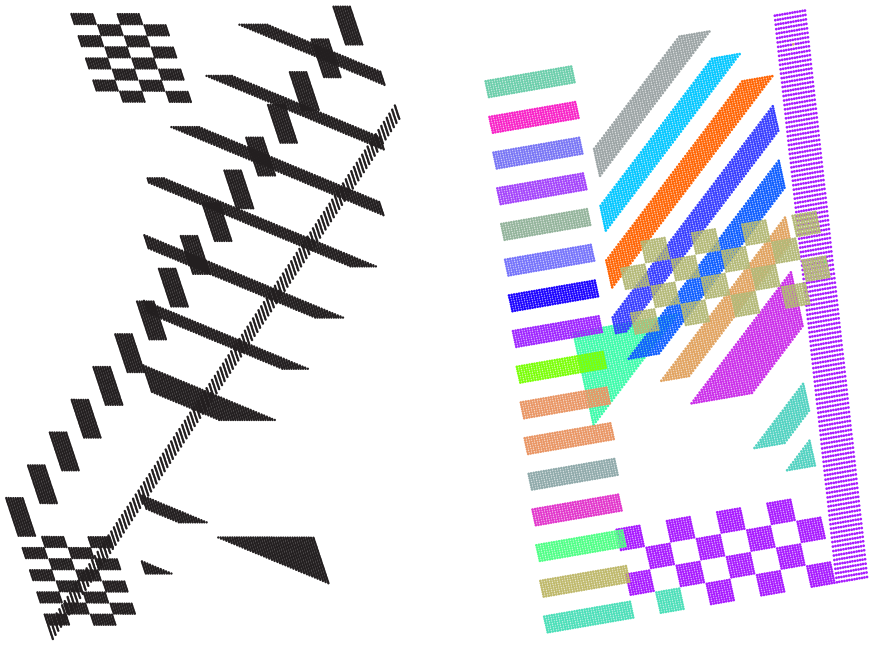
homogene Einteilung erwartete. Unserer Ansicht nach liefern der Modus 2 und 6 die besten Ergebnisse, wobei der Modus 6 auch den homogenen unteren Graukeil als eigenen Cluster identifiziert und nicht wie Modus 2 mit den dunklen Schachbrettfeldern vereinigt (Abb. 8.16). In Abbildung 8.17b ist eine dreidimensionale Ansicht des Ergebnisses von Modus 2 zu sehen. Es ist klar zu erkennen, wie die hellen und dunklen Schachbrettfelder voneinander getrennt wurden. Eine wesentlich stärkere Segmentierung liefern die Modi 3, 4, 7 und 8. Sie unterteilen auch den unteren Graukeil und fassen ihn nicht als einen Cluster zusammen. Es entstehen jedoch auch gleichzeitig (besonders im unteren Graukeil) vermehrt kleine punkt- und linienförmige Cluster, die subjektiv eher als Rauschen wahrgenommen werden und deshalb stören. Solche Störungen treten in den Modi 1 und 2 weniger auf, dafür ist jedoch der Graukeil in größere Gebiete aufgeteilt worden.

8.4 Ergebnisse für die realen Testdaten

8.4.1 Panchromatisches Luftbild

Bei der panchromatischen Luftaufnahme zeigt sich, wie im Falle des künstlichen Testbildes, dass die besten Ergebnisse die Modi 2 und 6 liefern (Abb. 8.18a, c). Die Maximierung des äußeren Abstands bewirkt hier, im Gegensatz zum künstlichen Testbild, eine vollständige Übersegmentierung (Abb. 8.18d), aber auch die Anwendung des HPGCL-Algorithmus ohne Iteration und Abstands-Maximierung liefert in diesem Fall keine zufriedenstellenden Ergebnisse, was anhand des Modus 5 (Abb. 8.18b) verdeutlicht werden soll.

Diese Ergebnisse müssen jedoch relativiert werden, denn die Abbildung des panchromatischen Luftbildes in den $x, y - Grauwert$-Raum dient nur als Beispiel für eine multispektrale Klassifikation und für den Merkmalsraum aus Lage und Grauwert sind natürlich nicht besonders vernünftige Ergebnisse zu erwarten. Umso erstaunlicher ist, dass im Modus 2 von den fünf Gebäuden in der Mitte der Luftaufnahme vier relativ gut wiedererkannt wurden (Abb. 8.18a). Das Straßennetz wurde ebenfalls teilweise wiedererkannt. Für eine klassische flächenhafte Bildsegmentierung ist unser Verfahren im allgemeinen nicht gedacht.



(a) Grauwerte des Testbilds im rgb-Raum (b) 3D Ansicht der gefundenen Cluster

Abbildung 8.17: Testbild als 3D-Punktmenge

8.4.2 3D-Laserdaten (Abstandsdaten)

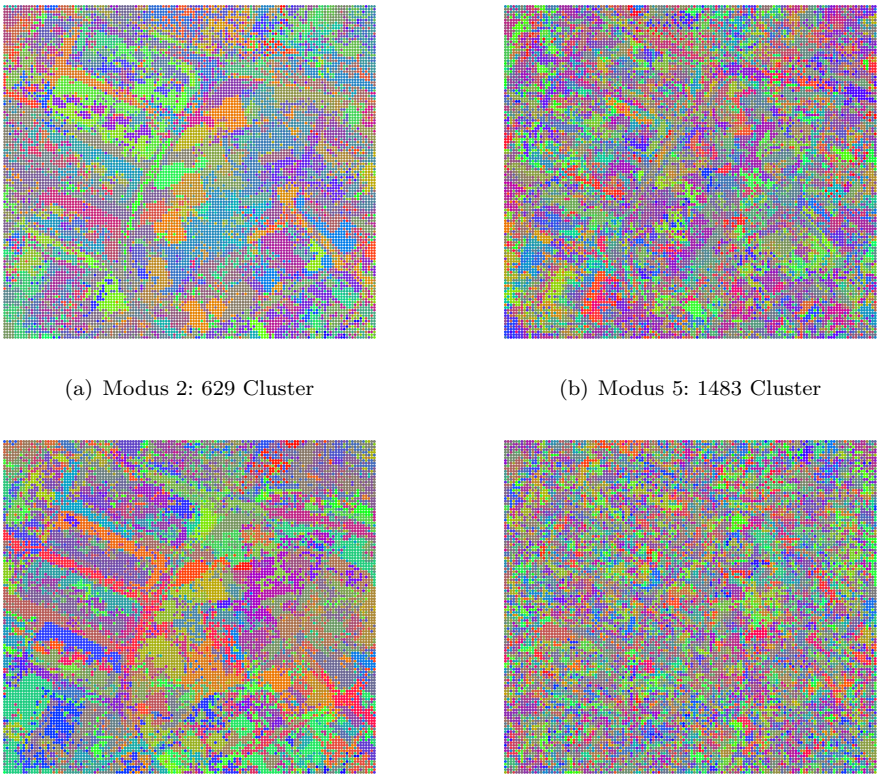
Die eingeschränkte Verwendbarkeit der Vergrößerung der äußeren Abstände (Modus 3, 4, 7 und 8) aufgrund der starken Übersegmentierung zeigen auch die Ergebnisse für die 3D-Abstandsdaten (siehe Abb. 8.19). Überraschend gute Ergebnisse ergeben die Modi 2 und 6, wenn man berücksichtigt, dass kein explizites Flächenmodell (z.B. Krümmungsverhalten) vorliegt. Auch die feineren Zerlegungen der Modi 1 und 5 sind erwartungsgemäß, wenn man berücksichtigt, dass die Laserdaten, wie in Abbildung 8.5 zu sehen, mehr oder weniger große Lücken aufweisen.

8.4.3 Gebäudedatensatz

Die für die manuelle Clusterauswertung verwendeten Punktmengen (Abb. 8.1) sind zwei Teilmengen der in Abb. 8.3 dargestellten Punktmenge. Die Ergebnisse des HPGCL-Algorithmus zu diesen Testdaten für Modus 1 bis 8 sind in den Abbildungen 8.20 und 8.21 dargestellt. Wie diese Abbildungen zeigen, liefern auch hier die Modi 2 und 6 die kompaktesten Zerlegungen, die auch gut vergleichbar mit den manuellen Auswertungen (siehe Anhang A) sind. Die automatischen Ergebnisse für den ersten Testdatensatz stimmen relativ gut mit den manuellen Ergebnissen überein. Für den zweiten Testdatensatz ergibt sich besonders für das Ergebnis aus Abbildung 8.21(b) eine relativ gute Übereinstimmung mit den manuellen Auswertungen in den Abbildungen A.2(a), A.3(a)(d) und A.4(b).

Der zweite Testdatensatz zeigt ebenfalls, dass die Segmentierung einer Punktmenge auch für den Menschen nicht eindeutig ist, da die Segmentierungsergebnisse hier stärker voneinander abweichen als im ersten Testdatensatz. Es zeigt sich jedoch, dass der HPGCL-Algorithmus generell stärker segmentiert als die manuellen Auswerter, d.h. die Anzahl der Cluster ist höher. Bei den manuellen Auswertungen wurden im ersten Testdatensatz im Mittel zwischen 11 und 15 Cluster bestimmt. Dagegen lieferte der HPGCL-Algorithmus 19 Cluster im Modus 2 und 22 Cluster im Modus 6. Im Falle des zweiten Testdatensatzes ergaben sich bei den manuellen Auswertungen zwischen 11 und 22 Cluster. Die automatische Auswertung lieferte hier 49 Cluster im Modus 2 und 48 Cluster im Modus 6.

Besonders am zweiten Testdatensatz zeigt sich, dass bei der automatischen Segmentierung mehr linienförmige Cluster erkannt werden als bei den manuellen Auswertungen, die überwiegend aus flächenförmigen Clustern



(a) Modus 2: 629 Cluster

(b) Modus 5: 1483 Cluster

(c) Modus 6: 641 Cluster

(d) Modus 8: 4415 Cluster

Abbildung 8.18: Ergebnisse für das panchromatische Luftbild

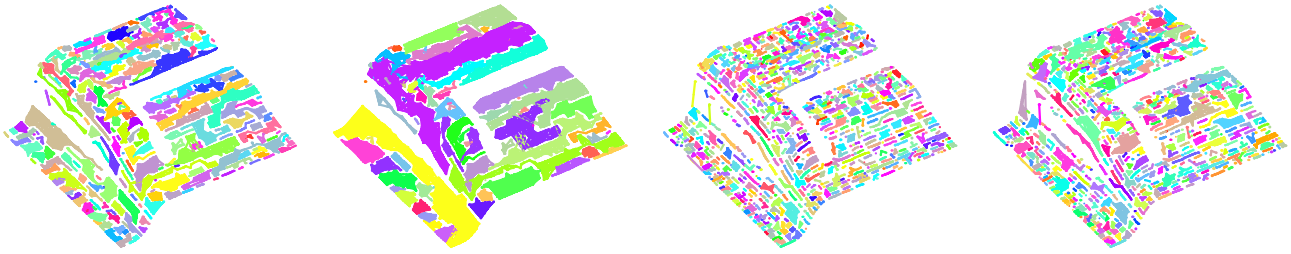
bestehen. Wie oben beschrieben zeigte der MST im iterativen Clustering ein besonders stark gruppierendes Verhalten, weswegen wir auch das Verhalten des HPGCL-Algorithmus für die einzelnen Graphen und die Teilhierarchien im Modus 2 auf beiden Testdatensätzen untersuchten. Die Ergebnisse sind in den Abbildungen 8.22 und 8.23 dargestellt. Wie sich bestätigt, erzeugen der MST, der RNG und die Teilhierarchie NNG – RNG überwiegend flächenförmige Cluster, die mit den manuellen Ergebnissen vergleichbar sind. Die Teilhierarchie NNG – MST liefert jeweils nur einen einzigen großen Cluster. In Anhang D sind die wesentlichen Ergebnisse der automatischen Auswertung des Vaihingen-Datensatzes für den iterativen HPGCL-Algorithmus zu sehen.

8.5 Laufzeitverhalten

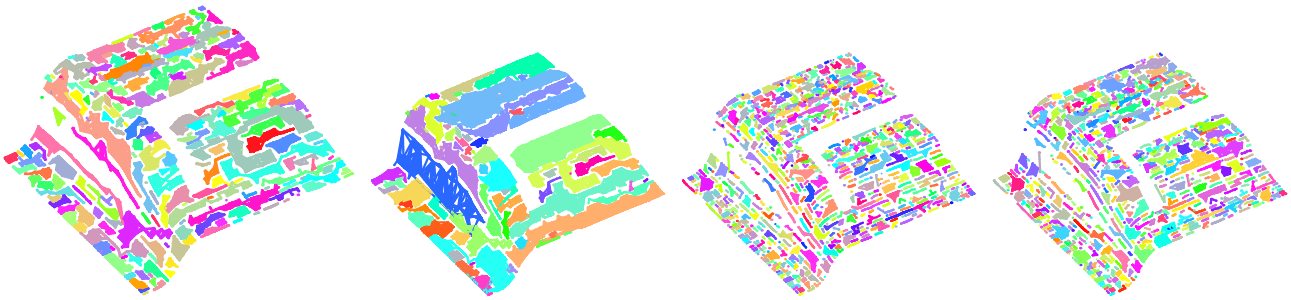
Auf die Zeitkomplexität für die Berechnung der Nachbarschaftsgraphen wurde schon im Kapitel 6 eingegangen. Nun wollen wir noch eine Abschätzung für die Zeitkomplexität der Clusterbildung des HPGCL-Algorithmus geben. Die Auswertung unserer Zeitmessungen (siehe Anhang B) zeigt, dass ein Laufzeitverhalten von $O(n^2)$ zu erwarten ist. In Abbildung 8.24 ist das Laufzeitverhalten für unsere Tests in doppelt-logarithmischer Darstellung abgebildet, wobei für die obere Schranke $o(n) = cn^2$ und die untere Schranke $u(n) = cn$ der konstante Faktor $c = 4 \cdot 10^{-5}$ gewählt wurde.

8.5.1 Diskussion

Als Schlussfolgerung kann man sagen, dass für eine allgemeine unüberwachte Klassifizierung sich die Modi 2 und 6 gut eignen. Die Modi 1 und 5 sollten verwendet werden, wenn eine detaillierte Segmentierung gewünscht ist. Die Modi 3, 4, 7 und 8 sind dagegen bei verrauschten Daten anzuwenden. Wird die Teilhierarchie NNG – MST angewendet, so sollte man die Modi 4 und 8 verwenden, um zu verhindern, dass nur ein einziger großer Cluster



(a) Modus 1: 239 Cluster (b) Modus 2: 73 Cluster (c) Modus 3: 899 Cluster (d) Modus 4: 738 Cluster



(e) Modus 5: 213 Cluster (f) Modus 6: 76 Cluster (g) Modus 7: 851 Cluster (h) Modus 8: 738 Cluster

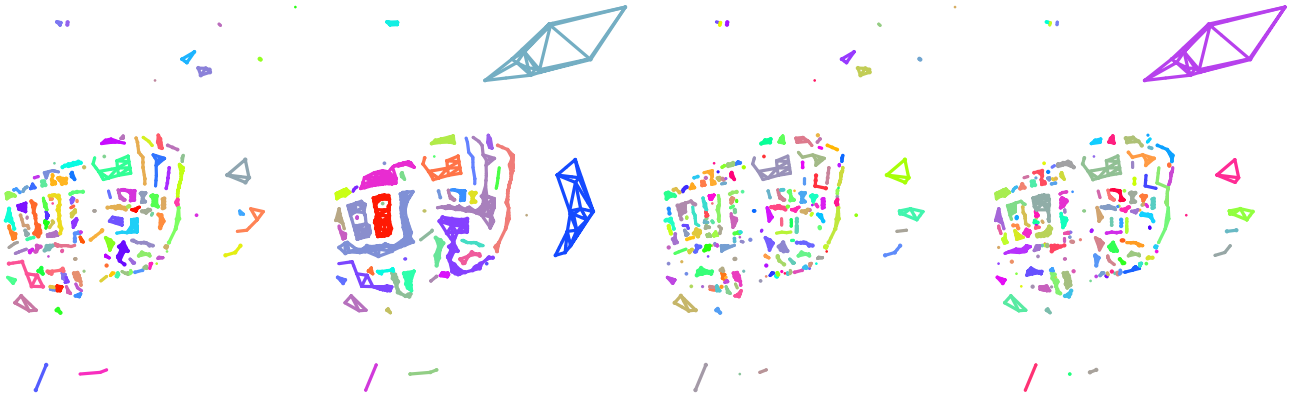
Abbildung 8.19: Ergebnisse für die 3D-Abstandsdaten

(a) Modus 1: 51 Cluster (b) Modus 2: 19 Cluster (c) Modus 3: 85 Cluster (d) Modus 4: 72 Cluster

(e) Modus 5: 51 Cluster (f) Modus 6: 22 Cluster (g) Modus 7: 89 Cluster (h) Modus 8: 72 Cluster

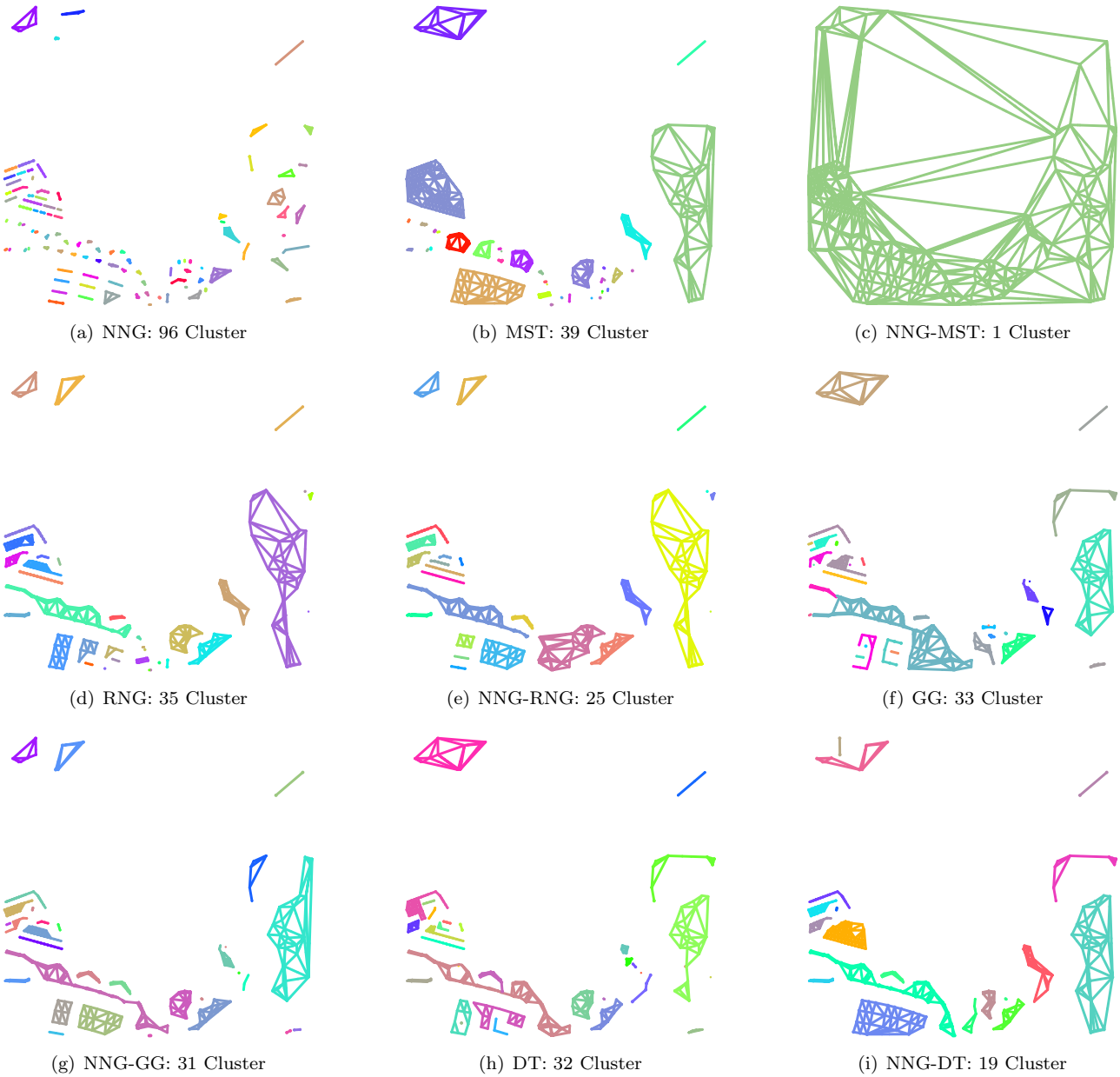
Abbildung 8.20: Automatische Auswertung des Testdatensatzes 1 (Modus 1-8)

(a) Modus 1: 127 Cluster (b) Modus 2: 49 Cluster (c) Modus 3: 192 Cluster (d) Modus 4: 159 Cluster



(e) Modus 5: 127 Cluster (f) Modus 6: 48 Cluster (g) Modus 7: 182 Cluster (h) Modus 8: 148 Cluster

Abbildung 8.21: Automatische Auswertung des Testdatensatzes 2 (Modus 1-8)



(a) NNG: 96 Cluster (b) MST: 39 Cluster (c) NNG-MST: 1 Cluster

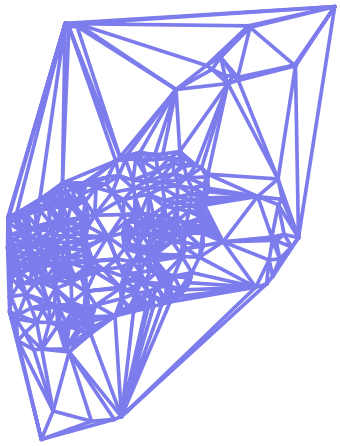
(d) RNG: 35 Cluster (e) NNG-RNG: 25 Cluster (f) GG: 33 Cluster

(g) NNG-GG: 31 Cluster (h) DT: 32 Cluster (i) NNG-DT: 19 Cluster

Abbildung 8.22: Ergebnisse für den Testdatensatz 1 bei Anwendung der einzelnen Graphen und Teilhierarchien im Modus 2.

(a) NNG: 188 Cluster

(b) MST: 33 Cluster



(c) NNG-MST: 1 Cluster

(d) RNG: 34 Cluster

(e) NNG-RNG: 44 Cluster

(f) GG: 67 Cluster

(g) NNG-GG: 59 Cluster

(h) DT: 78 Cluster

(i) NNG-DT: 49 Cluster

Abbildung 8.23: Ergebnisse für den Testdatensatz 2 bei Anwendung der einzelnen Graphen und Teilhierarchien im Modus 2.

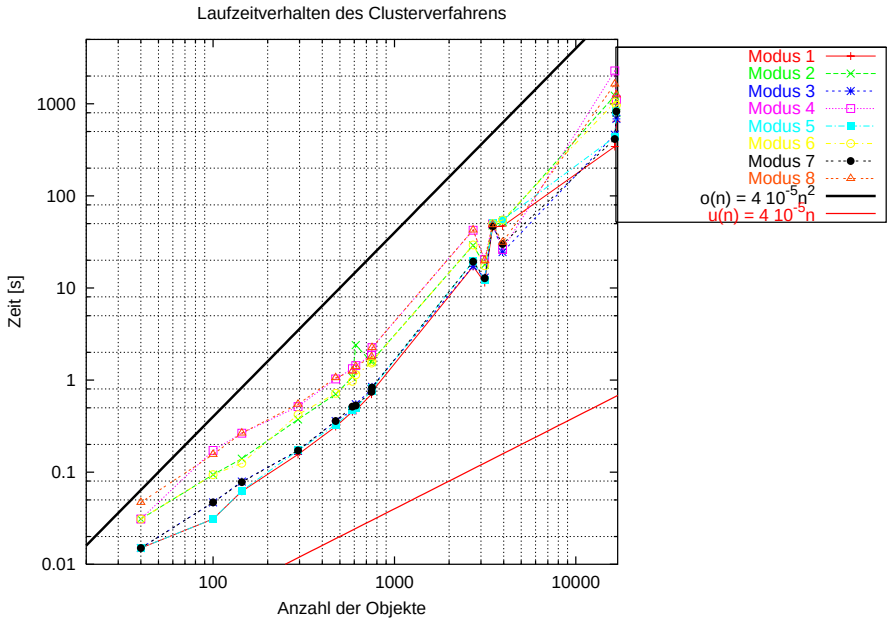
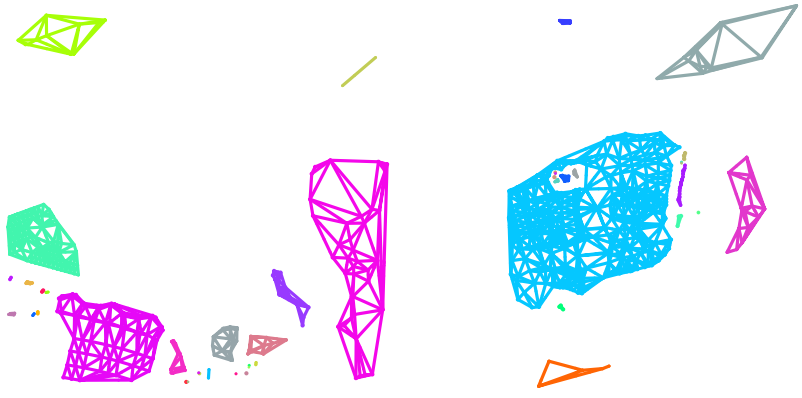


Abbildung 8.24: Laufzeitverhalten des Algorithmus für verschiedene Modi in doppelt-logarithmischer Darstellung

gebildet wird (siehe Abbildungen 8.22c und 8.23c). Für den zweiten Testdatensatz ergeben sich in diesem Fall die in Abbildung 8.25 dargestellten Ergebnisse. Generell sind für die Ableitung grober Segmentierungen der MST im Modus 2 und 6 (Abbildung 8.26 zeigt ein Beispiel für den Modus 6) sowie der RNG im Modus 2 zu empfehlen.

(a) Modus 4: 113 Cluster (b) Modus 8: 103 Cluster

Abbildung 8.25: Ergebnisse des iterativen HPGCL-Algorithmus für die Modi 4 und 8 bei Verwendung der Teilhierarchie NNG-MST.



(a) Testdatensatz 1 (b) Testdatensatz 2

Abbildung 8.26: Ergebnisse des iterativen HPGCL-Algorithmus für den Testdatensatz 1 und 2 im Modus 6 bei Verwendung des MST.

Kapitel 9

Diskussion und Ausblick

9.1 Zusammenfassung und Beurteilung

In dieser Arbeit befassen wir uns mit der automatischen Interpretation raumbezogener Daten. Aus dem breiten Spektrum der sich daraus ergebenden Problemstellungen konzentrieren wir uns auf das Problem der Aggregation von Objekten, das allgemein unter dem Begriff *räumliches Clustering* bekannt ist. Anhand von zwei Beispielen zeigen wir die Notwendigkeit solcher Verfahren für die Bildinterpretation und die kartographische Generalisierung auf.

Allen Cluster-Verfahren ist gemein, dass sie eine geeignete Definition der Nachbarschaft und der Ähnlichkeit von Objekten benötigen. In heutigen Standardmethoden wird dazu die Angabe von Parametern, wie z.B. Schwellwerte oder maximale Clusteranzahl, benötigt. Die geeignete Verwendung bestimmter Nachbarschaftsgraphen und ihrer hierarchischen Beziehung untereinander ermöglicht es uns, ein vollständig parameterfreies Verfahren zu beschreiben.

Im einzelnen zeigen wir in dieser Arbeit die folgenden Ergebnisse auf:

- Amtliche Katasterdaten (ALK) und Gebäudeinformationssysteme lassen sich hervorragend zur Ableitung von Vorinformationen für die wissensbasierte Bildinterpretation nutzen. Darüber hinaus zeigen wir, dass zweidimensionale räumliche Datenbanken mit Hilfe von Wahrscheinlichkeitsmodellen auch sehr gut zur approximativen, dreidimensionalen Visualisierung von Landschaftsmodellen genutzt werden können.
- Die automatische Ableitung von ATKIS-Daten aus ALK-Daten ist grundsätzlich möglich, was eine effizientere Erfassung von Basisinformationssystemen unterschiedlicher Maßstäbe ermöglicht.
- Nachbarschaftsgraphen eignen sich hervorragend zur Modellierung der Struktur räumlich verteilter, punkthafter Objekte. Das von uns vorgestellte Clustermodell baut auf diesen Nachbarschaftsgraphen auf und modelliert jeden Cluster als eine Menge von inneren Kanten und eine Menge von äußeren Kanten dieser Graphen. Die innere Kantenmenge sollte dabei so homogen wie möglich bezüglich der Kantenlänge sein. Der Median dieser Kantenlängen dient uns dabei als Schätzwert für den Erwartungswert der Kantenlänge und die absolute Medianabweichung (MAD) aller inneren und äußeren Kantenlängen stellt unser einfaches Unschärfemodell dar. Der Median und die MAD bilden die Basis für unser *unscharfes* Ähnlichkeitsmodell. Tests auf verschiedenen Datensätzen bestätigten die Eignung unseres einfachen Modells und die Allgemeinheit unseres Verfahrens bezüglich der Problemdomäne (Multispektraldaten, Laserdaten, GIS-Daten). Die Erweiterbarkeit unseres Verfahrens auf verschiedene Skalentypen und polyederförmige Objekte wird aufgezeigt.
- Die erhoffte generalisierende Wirkung (fein zu grob Segmentierung) der Hierarchie der Nachbarschaftsgraphen konnten anhand von unseren Tests bestätigt werden. Ebenfalls zeigte es sich, dass unser Verfahren auch robust gegenüber Rauschen ist, ohne ein explizites Rauschmodell anzuwenden.
- Die Tests ergaben ebenfalls, dass beim hierarchischen Clustering (wiederholte Gruppierung gefundener Cluster) unser Verfahren eine, bezüglich unserem Modell, eindeutige Grenzzerlegung aufweist, die im allgemeinen vom *Supercluster* des Objektraums (ein einziger Cluster aller Objekte) verschieden ist. Dies

unterscheidet unser Verfahren wesentlich von allgemeinen hierarchischen Clusterverfahren, die typischerweise immer einen einzigen Cluster als Grenzwert liefern und deshalb eine Vorgabe der maximalen Anzahl der Cluster benötigen.

- Unser Verfahren liefert Ergebnisse, die sehr gut mit der menschlichen Wahrnehmung vergleichbar sind und benötigt dafür keinerlei Vorgaben von Parametern, noch müssen solche geschätzt werden. Ebenfalls muss keinerlei Abbruchkriterium angegeben werden. Alle relevanten Informationen können aus den Nachbarschaftsgraphen abgeleitet werden. Somit bietet sich unser Verfahren hervorragend für die kartographische Generalisierung und für die explorative Datenanalyse an.

- Die Verwendung der Delaunay-Triangulation und die äußere und innere Kantenmenge liefert uns sofort eine geometrische Randbeschreibung der gefundenen Cluster.

9.2 Ausblick

Die von uns durchgeführten Untersuchungen deckten natürlich einige Probleme auf und ergaben folgende Fragestellungen für zukünftige Untersuchungen:

- Die Ableitung von Gebäudehypothesen aus Katasterdaten bringt die Frage auf, ob es generell möglich ist, durch Erfassung geeigneter, nicht geometrischer Attribute eine dreidimensionale Darstellung zu erzeugen, die beim Betrachter eine hohe Wiedererkennungsrate zur Realität erzeugt, ohne dass explizit die genaue dreidimensionale Form gespeichert wurde. Beispiele für solche Attribute wären z.B. der Fassadentyp, -material und -farbe oder ob die Dachform dem Grundriss folgt oder nicht. Wichtig wären auch Informationen über die Farbe und das Material des Daches. Wir wollen diese Problematik unter dem Begriff *Rapid Visualization* einordnen und sehen hier erheblichen Forschungsbedarf, da unserer Meinung die digitale Speicherung und Verwaltung der vollständigen dreidimensionalen Baupläne noch bis in die nahe Zukunft unter ökonomischen Gesichtspunkten nicht realisierbar sein wird. Eine Untersuchung zu typischen Gebäudemerkmalen und ihrer Wahrscheinlichkeit sowie der gegenseitigen Abhängigkeit wurde in (Fischer 1997) durchgeführt. Jedoch kann hier kein Anspruch auf Allgemeinheit gestellt werden, da dazu das untersuchte Gebiet zu klein war und auch die Menge der untersuchten Merkmale sehr klein war. Die automatische Ableitung von solchen typischen Merkmalen und ihrer Abhängigkeit untereinander könnte durch eine Clusteranalyse (wie z.B. mit unserem Verfahren) großer Gebäudeinformationssysteme unterschiedlicher Landschaftsregionen automatisiert werden.

- In ATKIS werden Straßen als Linien erfasst[1] und in der ALK als Flächen abgelegt; genauer gesagt wird in der ALK das Flurstück, auf dem die Straße verläuft, erfasst. In unserem Verfahren zur Ableitung von ATKIS-Daten aus ALK-Daten sind wir jedoch nicht auf die geeignete ATKIS-Generalisierung der Flurstücksgrenzen eingegangen und haben uns auf die Ableitung neuer Flächen ohne Anpassung der Geometrie beschränkt. Wann diese Generalisierung durchgeführt wird, d.h. vor oder nach der Ableitung neuer Flächenobjekte, ist unerheblich. Es ist jedoch zu untersuchen, wie die Ableitung dieser generalisierten Geometrie geeignet durchgeführt werden kann. Daraus ergeben sich zwei mögliche Lösungsansätze:

  - Vorgabe eines generalisierten Straßennetzes, womit sich das Problem auf ein geeignetes Matchingverfahren der Flurstücksgrenzen auf das gegebene Straßennetz beschränkt oder
  - der Ableitung der Straßengeometrie aus den ALK-Daten durch geeignete Skelettierung der Straßenflurstücke und gleichzeitigem oder separatem Matching der gegebenen Flurstücksgrenzen.

  Generell wäre es interessant zu wissen, wie genau man die ATKIS-Straßengeometrie aus ALK-Daten ableiten kann.

- Die Auswertung der manuell erfassten Cluster zeigte, dass der Mensch bei der Gruppierung von punkthaften Objekten nicht nur die homogene Anordnung berücksichtigt, sondern auch die Form des Clusters. Die Erweiterung unseres Verfahrens um Formeigenschaften, wie z.B. linear, kreisförmig oder rechteckig wäre eine interessante Aufgabe, die das Ergebnis unseres Verfahrens sicherlich noch weiter verbessern würde.

[1] Ausnahme sind befahrbare Plätze, die als Fläche erfasst werden.

- Neben der expliziten Berücksichtigung der Form eines Clusters ist auch die Definition und Verwendung von Rauschmodellen eine offene Frage und bietet ein breites Feld an Forschungsmöglichkeiten.

- Auch wenn es unser Ziel war, ein vollständig parameterfreies Verfahren zu definieren, so wäre die Angabe eines Qualitätsmaßes wünschenswert, wie z.B. *finde die Gruppen von Objekten, die mit 90% Wahrscheinlichkeit zusammengehören.* Eine andere Möglichkeit wäre, dass das Verfahren zu jedem Cluster einen Wahrscheinlichkeitswert oder ein Konfidenzintervall liefert, denn unser Verfahren liefert nur den MAD (Homogenität) der Clusterdichte als Qualitätsmaß eines Clusters. Um eine fundierte qualitative Beurteilung zu ermöglichen, müsste untersucht werden, wie unser Verfahren mit einem verteilungsunabhängigen Testverfahren (parameterfreie Statistik) erweitert werden kann, denn eine Annahme über die Art der Verteilung sollte weiterhin nicht notwendig sein.

- Das Fundament unseres beschriebenen Verfahrens ist die Delaunay-Triangulation, da aus ihr in effizienter Weise alle anderen Nachbarschaftsgraphen abgeleitet werden können und sie es uns zusätzlich ermöglicht, mit Hilfe der äußeren und inneren Kantenmengen eine Randbeschreibung der gefundenen Cluster angeben zu können. So schön dieser formale Ansatz auch ist, so hat er in der Praxis jedoch einen wesentlichen Nachteil. Im Falle großer und hochdimensionaler ($d > 4$) Datensätze ist die Delaunay-Triangulation im allgemeinen nicht geeignet, denn die Zeitkomplexität zur Berechnung der Delaunay-Triangulation beträgt für $d > 2$ dann $O(n^{\lceil \frac{d}{2} \rceil})$, gegenüber $O(nlogn)$ im zweidimensionalen Fall, und die Speicherkomplexität ist dann ebenfalls nicht mehr linear sondern $O(n^2)$. Der *relative Nachbarschaftsgraph* und der *Gabriel-Graph* weisen zwar bei direkter Berechnung eine etwas günstigere Zeitkomplexität auf, sie besitzen jedoch ebenfalls für $d > 3$ keine lineare Speicherkomplexität mehr. In diesen Fällen erweisen sich die *k-Nächsten-Nachbargraphen* und der *minimal spannende Baum* als erheblich günstiger, da sie im allgemeinen für beliebige Dimensionen eine Speicherkomplexität von $O(n)$ besitzen. In (Kleinberg 1997) und (Eppstein 1998) werden effiziente Methoden zur Berechnung von Nächsten-Nachbargraphen für beliebige Dimensionen beschrieben. Es wäre deshalb sehr interessant zu untersuchen, wie sich die Hierarchie der k-Nächsten-Nachbargraphen verhält und ob unser Verfahren mit diesen Graphen ähnliche oder vielleicht sogar bessere Ergebnisse liefert. Welche k-Nächsten-Nachbargraphen wären notwendig (z.B. 1, 2, 3, 4 oder 5)? Verhält sich diese Hierarchie bezüglich der Clusteranzahl in gleicher Weise, wie die von uns verwendete Hierarchie?

- Als letztes sei noch unser iterativer Clusteransatz erwähnt. Das von uns definierte Modell wurde eingesetzt, da es eine einzige einheitliche Clusterdefinition und Aggregationsvorschrift für den nicht-iterativen und iterativen Fall ermöglicht und somit nicht zwischen *Clustern von Clustern* und *Clustern von einzelnen Objekten* unterschieden werden muss. Bei diesem Modell gehen jedoch alle gewonnenen Informationen über ein Cluster bei jedem Iterationsschritt verloren, was besonders bei regelmäßig angeordneten Strukturen zur Gruppierung benachbarter Cluster unterschiedlicher Dichte führt (siehe Testergebnisse für die künstlichen Testdaten in Abbildung 8.13 auf Seite 97). Es stellt sich somit die Frage, ob, und wenn ja wie, unser Modell erweitert werden kann, um ein einheitliches Modell zu erhalten, das nur die iterative Gruppierung von regelmäßig angeordneten Clustern gleicher Dichte erlaubt und Cluster unterschiedlicher Dichte verbietet.

Literaturverzeichnis

Agarwal, P. & Matousek, J. (1992), Relative neighborhood graphs in three dimension, *in:* '3rd Annual ACM Symposium Discrete Algorithms', Seiten 58–67.

Agrawal, R. & Srikant, R. (1994), Fast Algorithms for Mining Association Rules, *in:* J. B. Bocca, M. Jarke & C. Zaniolo, Hrsg., 'Proc. 20th Int. Conf. Very Large Data Bases, VLDB', Morgan Kaufmann, Seiten 487–499.

Agrawal, R., Imielinski, T. & Swami, A. N. (1993), Mining Association Rules between Sets of Items in Large Databases, *in:* P. Buneman & S. Jajodia, Hrsg., 'Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data', Washington, D.C., Seiten 207–216.

Aigner, M. (1984), *Graphentheorie: Eine Entwicklung aus dem 4-Farben Problem*, Teubner, Stuttgart. ISBN 3-519-02068-8.

Anand, S., Bell, D. & Hughes, J. (1993), 'A general framework for database mining based on evidential theory', Internal Report, Dept. of Inf.Sys., University of Ulster at Jordanstown.

Anders, K.-H. (1997), Automated interpretation of digital landscape models, *in:* 'Photogrammetric Week '97', Wichmann Verlag - Heidelberg, Seiten 13–24.

Anders, K.-H. (2001), Data Mining for Automated GIS Data Collection, *in:* 'Photogrammetric Week '01', Wichmann Verlag - Heidelberg, Seiten 263–272.

Anders, K.-H. & Sester, M. (1997), Methods of Data Base Interpretation - Applied to Model Generalization from Large to Medium Scale, *in:* W. Förstner & L. Plümer, Hrsg., 'Semantic Modeling for the Acquisition of Topographic Information from Images and Maps / SMATI 97', Birkhäuser, Bonn, Germany, Seiten 89–103.

Anders, K.-H. & Sester, M. (2000), Parameter-Free Cluster Detection in Spatial Databases and ist Application to Typification, *in:* 'International Archives of Photogrammetry and Remote Sensing', Vol. 33(Part B4/1), ISPRS Congress, Amsterdam, Seiten 75–82. Comm. IV.

Anders, K.-H., Sester, M. & Fritsch, D. (1997), Automation of Spatial Analysis Methods, *in:* Y. Lee & Z. li Li, Hrsg., 'International Workshop on Dynamic and Multi-Dimensional GIS', Advanced Geographic Data Modelling, ISPRS, Inter-Commission Working Group IV/III.1, Departement of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, Hong Kong.

Anders, K.-H., Sester, M. & Fritsch, D. (1999), Analysis of Settlement Structures by Graph-Based Clustering, *in:* W. Förstner, C.-E. Liedtke & J. Bückner, Hrsg., 'SMATI 99: Semantic Modelling for the Acquisition of Topographic Information from Images and Maps', Munich, Germany, Seiten 41–49.

Arslan, A. N. & Egecioglu, O. (2000), 'Efficient Algorithms for Normalized Edit Distance', *J. Discret. Algorithms* **1**(1), 3–20.

Bajcsy, P. & Ahuja, N. (1998), 'Location- and Density-Based Hierarchical Clustering Using Similarity Analysis', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(9), 1011–1015.

Ball, G. & Hall, D. (1965), 'ISODATA: a novel method of data analysis and pattern classification', Stanford Research Institute AD 699616.

Baltsavias, E., Grün, A. & Van Gool, L. e., Hrsg. (2001), *Automatic Extraction of Man-Made Objects from Aerial and Space Images (III)*, Balkema Publishers, Rotterdam.

Bell, D., Anand, S. & Shapcott, C. (1994), 'Database Mining in Spatial Databases', International Workshop on Spatio-Temporal Databases.

Bellman, R. (1957), *Dynamic Programming*, Princeton University Press.

Berkhin, P. (2002), 'Survey Of Clustering Data Mining Techniques', Accure Research Papers (http://www.accrue.com/products/researchpapers.html).

Bill, R. & Fritsch, D. (1991), *Grundlagen der Geo-Informationssysteme: Hardware, Software und Daten*, Wichmann, Karlsruhe.

Brenner, C. (2000), Dreidimensionale Gebäuderekonstruktion aus Oberflächenmodellen und Grundrissen, Doktorarbeit, Universität Stuttgart, Deutsche Geodätische Kommission, Reihe C, Nr. 530, München.

Brinkhoff, T., Kriegel, H.-P. & Seeger, B. (1993), Efficient Processing of Spatial Joins using R-trees, *in:* 'Int. Conf. on Management of Data', ACM-SIGMOD, Washington, D.C., Seiten 237–246.

Damerau, F. (1964), 'A Technique fir Computer Detection and Correction of Spelling Errors', *Communications of the ACM* **7**, 171–176.

Diestel, R. (2000), *Graphentheorie, 2. Auflage*, Springer-Verlag, Heidelberg. ISBN 3-540-67656-2.

Egenhofer, M. & Franzosa, R. (1995), 'On the equivalence of topological relations', *International Journal of GIS* **9**(2), 133–152.

Eppstein (1998), Fast Hierarchical Clustering and Other Applications of Dynamic Closest Pairs, *in:* 'SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)'.

Eppstein, Paterson & Yao (1997), 'On Nearest Neighbor Graphs', *GEOMETRY: Discrete & Computational Geometry*.

Ester, M., Kriegel, H.-P. & Xu, X. (1995), Knowledge Discovery in Large Spatial Databases: Focusing Techniques for Efficient Class Identification, *in:* 'Advances in Spatial Databases (Proc. 4th Symp. SSD-95)', Portland, ME, Seiten 67–82.

Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. (1996), A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *in:* 'Proceedings of 2nd. International Conference on Knowledge Discovery and Data Mining (KDD-96)'.

Estivill-Castro, V. & Lee, I. (2002), 'Multilevel Clustering and its Visualization for Exploratory Spatial Analysis', *GeoInformatica* **6**(2), 123–152.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy, R., Hrsg. (1996), *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT, Menlo Park, CA.

Fischer, J. (1997), 'Analyse von Gebäudemerkmalen am Beispiel Stuttgart-Berg'.

Förstner, W. (1999), 3D-City Models: Automatic and Semiautomatic Acquisition Methods, *in:* 'Photogrammetric Week '99', Wichmann Verlag - Heidelberg, Seiten 291–303.

Fotheringham, S. & Rogerson, P. (1994), *Spatial Analysis and GIS*, Taylor and Francis.

Frawley, W., Piatetsky-Shapiro, G. & Matheus, C. (1991), Knowledge Discovery in Databases: An Overview, *in:* G. Piatetsky-Shapiro & W. Frawley, Hrsg., 'Knowledge Discovery in Databases', AAAI/MIT Press, Menlo Park, CA, Seiten 1–27.

Fritsch, D. & Anders, K.-H. (1996), 'Objektorientierte Konzepte in Geo-Informationssystemen', *GIS GEO-Informations-Systeme* **9**(2), 2–14.

Gabriel, K. & Sokal, R. (1969), 'A new statistical approach to geographic variation analysis', *Systematic Zoology* **18**, 259–278.

Goodman, J. & O'Rourke, J., Hrsg. (1997), *Handbook of Discrete and Computational Geometry*, CRC Press, Boca Raton, New York.

Goodrich, M., Tsay, J.-J., Vengroff, D. & Vitter, J. (1993), External-Memory Computational Geometry, *in:* '34th Symposium on Foundations of Computer Science'.

Guha, S., Rastogi, R. & Shim, K. (1998), CURE: An efficient clustering algorithm for large databases, *in:* 'Proc. of 1998 ACM-SIGMOD International Conference on Management of Data'.

Guha, S., Rastogi, R. & Shim, K. (1999), ROCK: A robust clustering algorithm for categorical attributes, *in:* 'Proc. of the 15th International Conference on Data Engineering'.

Güttman, R. (1984), A dynamic index structure for spatial searching, *in:* 'Int. Conf. on Management of Data', ACM-SIGMOD, Boston, MA, Seiten 47–57.

Haala, N. & Anders, K.-H. (1996), Fusion of 2D-GIS and Image Data for 3D Building Reconstruction, *in:* 'International Archives of Photogrammetry and Remote Sensing', Vol. 31(3), ISPRS, Vienna, Austria, Seiten 285–290.

Haala, N. & Anders, K.-H. (1997), Acquisition of 3D urban models by analysis of aerial images, digital surface models and existing 2D building information, *in:* 'SPIE Conference on Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision III', SPIE, Orlando, Florida, Seiten 212–222.

Hamming, R., Hrsg. (1987), *Information und Codierung*, VCH Verrlagsgesellschaft, Weinheim.

Han, J. & Fu, Y. (1996), Exploration of the Power of Attribute-Oriented Induction in Data Mining, *in:* U. Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy, Hrsg., 'Advances in Knowledge Discovery and Data Mining', AAAI/MIT, Menlo Park, CA.

Hand, D. (1981), *Discrimination and Clssification*, Wiley, Chichester.

Hein, O. (1977), *Graphentheorie für Anwender*, Vol. 83, Bibliographisches Institut AG, Mannheim. ISBN 3-411-00083-X.

Hermes, H., Hrsg. (1967), *Einführung in die Verbandstheorie, 2. Auflage*, Springer-Verlag, Berlin-Heidelberg.

Holsheimer, M. & Kersten, M. (1994), Architectural Support for Data Mining, Technical Report CS-R9429, CWI, Amsterdam, The Netherlands.

Holsheimer, M. & Siebes, A. (1994), Data mining: The search for knowledge in databases, Technical Report CS-R9406, CWI, Amsterdam, The Netherlands.

Jain, A. & Dubes, R. (1988), *Algorithms for Clustering Data*, Prentice Hall.

Jaromczyk, J. & Toussaint, G. (1992), Relative neighborhood graphs and their relatives, *in:* 'Proceedings IEEE', Vol. 80(9), Seiten 1502–1517.

Jarvis, R. & Patrick, E. (1973), 'Clustering using a similarity measure based on shared near neighbours', *IEEE Transactions on Computers* **22**(11), 1025–1034.

Karypis, G., Han, E.-H. S. & Kumar, V. (1999), CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamical Modeling, To appear in the IEEE Computer or via internet at http://winter.cs.umn.edu/∼karypis/publications/data-mining.html.

Kaufman, L. & Rousseeuw, P. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons.

Kaufmann, M. (1971), *Einführung in die Graphentheorie*, R. Oldenbourg, München. ISBN 3-486-33511-1.

King, V. (1995), A Simpler Minimum Spanning Tree Verification Algorithm, *in:* 'Workshop on Algorithms and Data Structures', Seiten 440–448.

Kirkpatrick, D. & Radke, J. (1985), A framework for computational morphology, *in:* G. Toussaint, Hrsg., 'Computational Geometry', North-Holland, Seiten 217–248.

Kleinberg, J. M. (1997), Two algorithms for nearest-neighbor search in high dimensions, *in:* 'Proc. 29th ACM Symposium on Theory of Computing,', Seiten 599–608.

Koperski, K., Adhikary, J. & Han, J. (1996), Knowledge Discovery in Spatial Databases: Progress and Challenges, *in:* 'Proceedings of Workshop on Research Issues on Data Mining and Knowledge Discovery', Montreal, QB.

Koperski, K. & Han, J. (1995), Discovery of Spatial Association Rules in Geographic Information Databases, *in:* 'Advances in Spatial Databases (Proc. 4th Symp. SSD'95)', Portland, ME, Seiten 47–66.

Kuhn, W. (2001), 'Ontologies in support of activities in geographical space', *International Journal of Geographical Information Science* **15**(7), 613–631.

Lee, D. (1980), 'Two dimensional voronoi diagram in the l_p metric', *Journal of ACM* (27), 604–618.

Levenshtein, V. (1965), 'Binary codes capable of correcting deletions, insertions and reversals', *Doklady Akademii Nauk SSSR* **4**(163), 845–848.

Lingas, A. (1994), 'A linear-time construction of the relative neighborhood graph from the Delaunay triangulation', *Computational Geometry* **4**(4), 199–208.

Lu, W., Han, J. & Ooi, B. (1993), Discovery of General Knowledge in Large Spatial Databases, *in:* 'Proc. of 1993 Far East Workshop on Geographic Information Systems (FEGIS'93)', Singapore, Seiten 275–289.

MacQueen, J. (1967), Some methods for classification and analysis of multivariate observations, *in:* 'Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability', Vol. 1, Seiten 281–297.

Matheus, C., Chan, P. & Piatetsky-Shapiro, G. (1993), 'Systems for Knowledge Discovery in Databases', *IEEE Transaction on Knowledge and Data Engeneering* **5**, 903–913.

McLachlan, G. (1992), *Discriminant Analysis and Statistical Pattern Recognition*, John Wiley & Sons, New York.

Michalski, R., Carbonell, J. & Mitchell, T. (1984), *Machine Learning - An Artificial Intelligence Approach*, Springer-Verlag, Berlin.

Mohan, R. & Nevatia, R. (1988), Perceptual Grouping for the Detection and Description of Structures in Aerial Images, *in:* 'Proceedings of the DAPRA Image Understanding Workshop', Cambridge, MA, Seiten 512–526.

Molenaar, M. (1996), The role of topologic and hierarchical spatial object models in database generalization, *in:* M. Molenaar, Hrsg., 'Methods for the Generalization of Geo-Databases', number 43, Netherlands Geodetic Commission, Delft, The Netherlands, Seiten 13–35.

Nakano, K. & Olariu, S. (1997), 'An Optimal Algorithm for the Angle-Restricted All Nearest Neighbor Problem on the Reconfigurable Mesh, with Applications:', *IEEE Transactions on Parallel and Distributed Systems* **8**(9), 983–990.

Ng, R. & Han, J. (1994), Efficient and Effective Clustering Method for Spatial Data Mining, *in:* 'Proc. of 1994 Int. Conf. on Very Large Data Bases (VLDB'94)', Santiago, Chile, Seiten 144–155.

O'Rourke, J. (1982), 'Computing the Relative Neighborhood Graph in the l_1 and l_∞ metrics', *Pattern Recognition* Seiten 45–55.

Ottmann, T. & Widmayer, P. (1993), *Algorithmen und Datenstrukturen*, BI Wissenschaftsverlag. 2. vollständig überarbeitete und erweiterte Auflage.

Pfeifer, U., Poersch, T. & Fuhr, N. (1995), Searching Proper Names in Databases, *in:* 'HIM', Seiten 259–275.

Piatetsky-Shapiro, G. & Frawley, W., Hrsg. (1991), *Knowledge Discovery in Databases*, AAAI/MIT Press, Menlo Park, CA.

Preparata, F. & Shamos, M. (1985), *Computational Geometry: An Introduction*, Springer-Verlag, New York.

Preparata, F. P. & Shamos, M. I. (1988), *Computational Geometry*, Springer-Verlag, New York.

Rao, S. (1998), Some Studies on Beta-Skeletons, Doktorarbeit, Department of Computer Science & Engineering, Indian Institute of Technology, Kanpur.

Regnauld, N. (1996), Recognition of Building Clusters for Generalization, *in:* M. Kraak & M. Molenaar, Hrsg., 'Advances in GIS Research, Proc. of 7th Int. Symposium on Spatial Data Handling (SDH)', Vol. 1, Faculty of Geod. Engineering, Delft, The Netherlands, Seiten 4B.1–4B.14.

Reinhardt, F. & Soeder, H. (1984), *dtv-Atlas zur Mathematik, Band I Grundlagen, Algebra und Geometrie*, Deutscher Taschenbuch Verlag, München. ISBN 3-423-03007-0.

Sachs, L. (1999), *Angewandte Statistik: Anwendung statistischer Methoden*, Springer-Verlag, Berlin Heidelberg. ISBN 3-540-65371-6.

Samet, H. (1990), *The Design and Analysis of Spatial Data Structures*, Addison-Wesley.

Sankoff, D. & Kruskal, J., Hrsg. (1983), *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparsion*, Addison-Wesley, Kanpur.

Sedlacek, J. (1971), *Einführung in die Graphentheorie*, B. G. Teubner Verlagsgesellschaft, Leipzig.

Sester, M. (1995), *Lernen struktureller Modelle für die Bildanalyse*, Vol. C441, Deutsche Geodätische Kommission, München.

Sester, M., Anders, K.-H. & Walter, V. (1998), 'Linking Objects of Different Spatial Data Sets by Integration and Aggregation', *GeoInformatica* **2**(4), 335–358.

Shaw, G. & Wheeler, D. (1994), *Statistical Techniques in Geographical Analysis*, David Fulton, London.

Sniedovich, M., Hrsg. (1992), *Dynamic Programming*, Marcel Dekker, New York.

Supowit, K. (1983), 'The relative neighborhood graph, with an application to minimum spanning trees', *J.Assoc.Comput.Mach.* **30**, 428–448.

Titterington, D., Smith, A. & Makov, U. (1985), *Statistical Analysis of Finite Mixture Distributions*, John Wiley & Sons, New York.

Tobler, W. (1970), 'A computer movie simulating urban growth in the Detroit region', *Economic Geography* **46**(2), 234–240.

Toussaint, G. (1980*a*), 'Algorithms for computation of relative neighbourhood graph', *Electronic Letters* **16**(22), 860–861.

Toussaint, G. (1980*b*), 'The relative neighborhood graph of a finite planar set', *Pattern Recognition* **12**, 261–268.

Toussaint, G. (1988), A graph-theoretical primal sketch, *in:* G. Toussaint, Hrsg., 'Computational Morphology', North-Holland, Seiten 229–260.

Toussaint, G. (1991), Some unsolved problems on proximity graphs, *in:* D. Dearholt & F. Harary, Hrsg., 'Proceedings of the First Workshop on Proximity Graphs. Memoranda in Computer and Cognitive Science MCCS-91-224', Computing research laboratory, New Mexico State University, La Cruces.

Ukkonen, E. (1985), 'Algorithms for approximate string matching', *Information and Control* **64**, 100–118.

Urquhart, R. (1980), 'Algorithms for computation of relative neighbourhood graph', *Electronic Letters* **16**(14), 556–557.

Urquhart, R. (1982), 'Graph theoretical clustering based on limited neighborhood sets', *Pattern Recognition* **15**, 173–187.

van Dongen, S. (2000), Graph Clustering by Flow Simulation, Doktorarbeit, Universität Utrecht, Center for Mathematics and Computer Science (CWI), Amsterdam.

van Schröder, M. (2001), Gebiete optimal aufteilen OR-Verfahren für die Gebietsaufteilung als Anwendungsfall der gleichmäsiger Baumzerlegung, Doktorarbeit, Universität Karlsruhe, Fakultät für Wirtschaftswissenschaften.

Walter, V. (1997), *Zuordnung von raumbezogenen Daten - am Beispiel ATKIS und GDF*, Dissertation, Deutsche Geodätische Kommission (DGK) Reihe C, Heft Nr. 480.

Weiss, S. M. & Indurkhya, N. (1998), *Predictive Data Mining a Practical Guide*, Morgan Kaufmann Publishers, Inc., San Francisco, California. ISBN 1-55860-403-0.

Weiss, S. M. & Kulikowski, C. (1991), *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*, Morgan Kaufmann Publishers, Inc., San Francisco, California.

Winter, S., Hrsg. (2001), *SPECIAL ISSUE Ontology in the Geographic Domain*, Vol. 15(7) of *International Journal of Geographical Information Science*, Taylor & Francis, London, Philadelphia.

Yao, A.-C. (1982), 'On constructing minimum spanning trees in k-dimensional spaces and related problems', *SIAM J. Comput.* (11), 721–736.

Zahn, C. (1971), 'Graph-theoretical methods for detecting and describing gestalt clusters', *IEEE Transactions on Computers* **C**(20), 68–86.

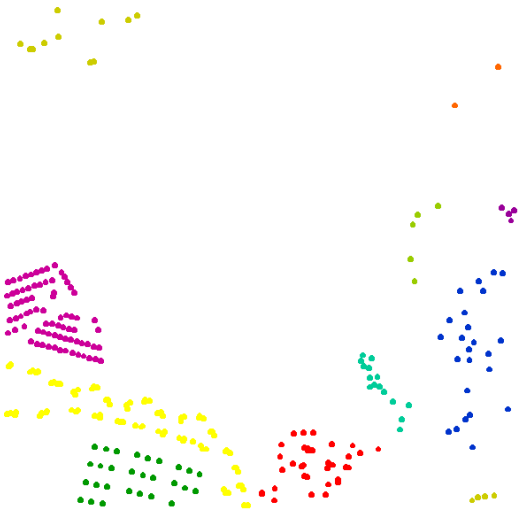
Zahn, M. (1996), Klassifizierung von multispektralen Bildern unter Verwendung von Clusterformen im Merkmalsraum, Doktorarbeit, Universität Karlsruhe, München.

Anhang A

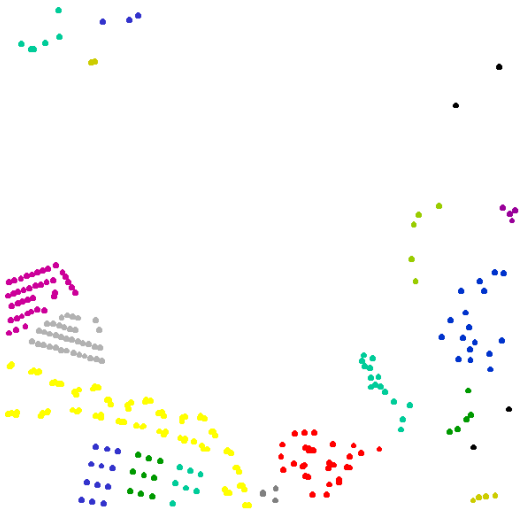
Manuelle Auswertungen von Testmuster 1 und 2

(a) Testmuster 1 (b) Testmuster 2

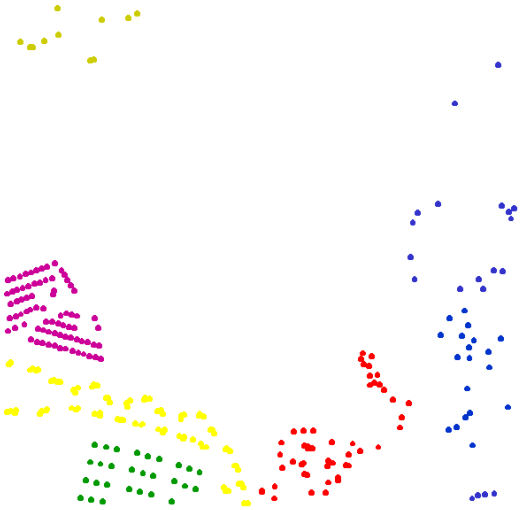
Abbildung A.1: Testmuster 1 und 2



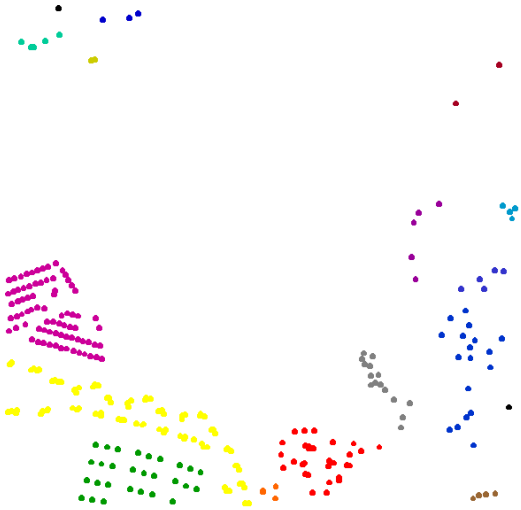
(a) Auswertung 1



(b) Auswertung 2

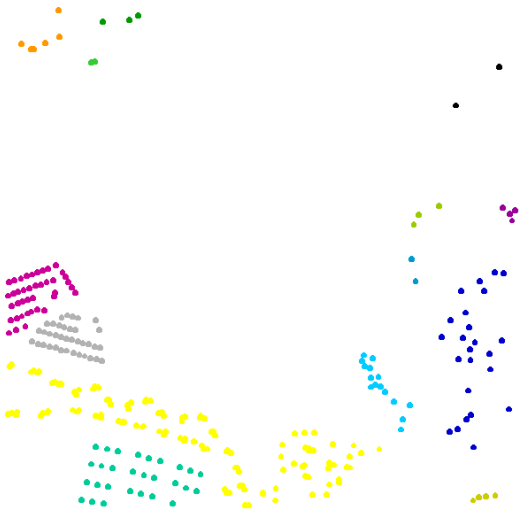


(c) Auswertung 3

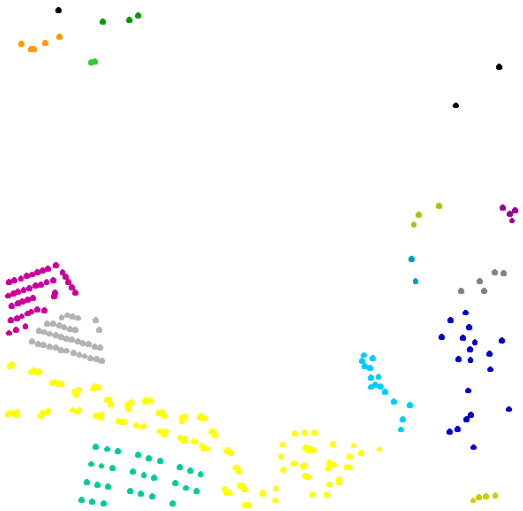


(d) Auswertung 4

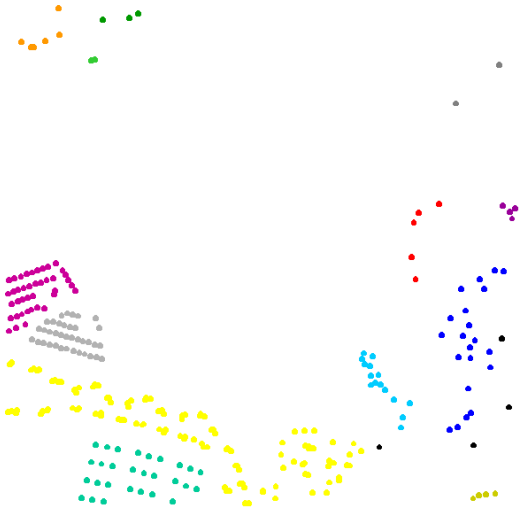
Abbildung A.2: Testmuster 1: Auswertungen 1 bis 4



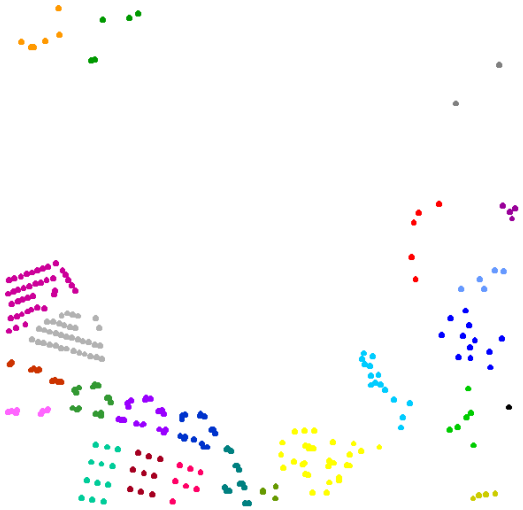
(a) Auswertung 5



(b) Auswertung 6

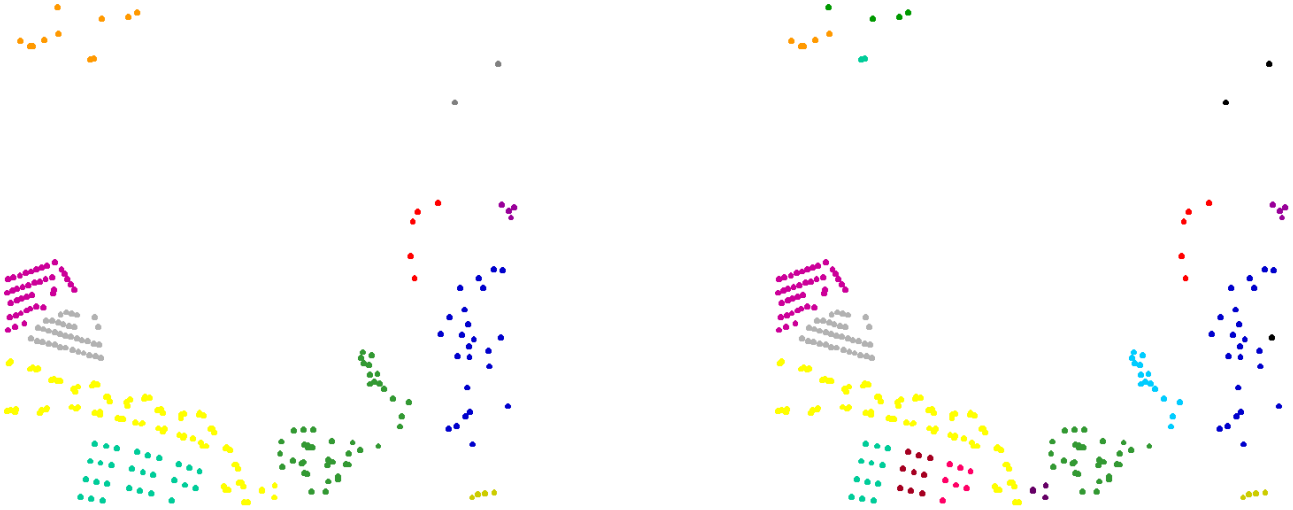


(c) Auswertung 7



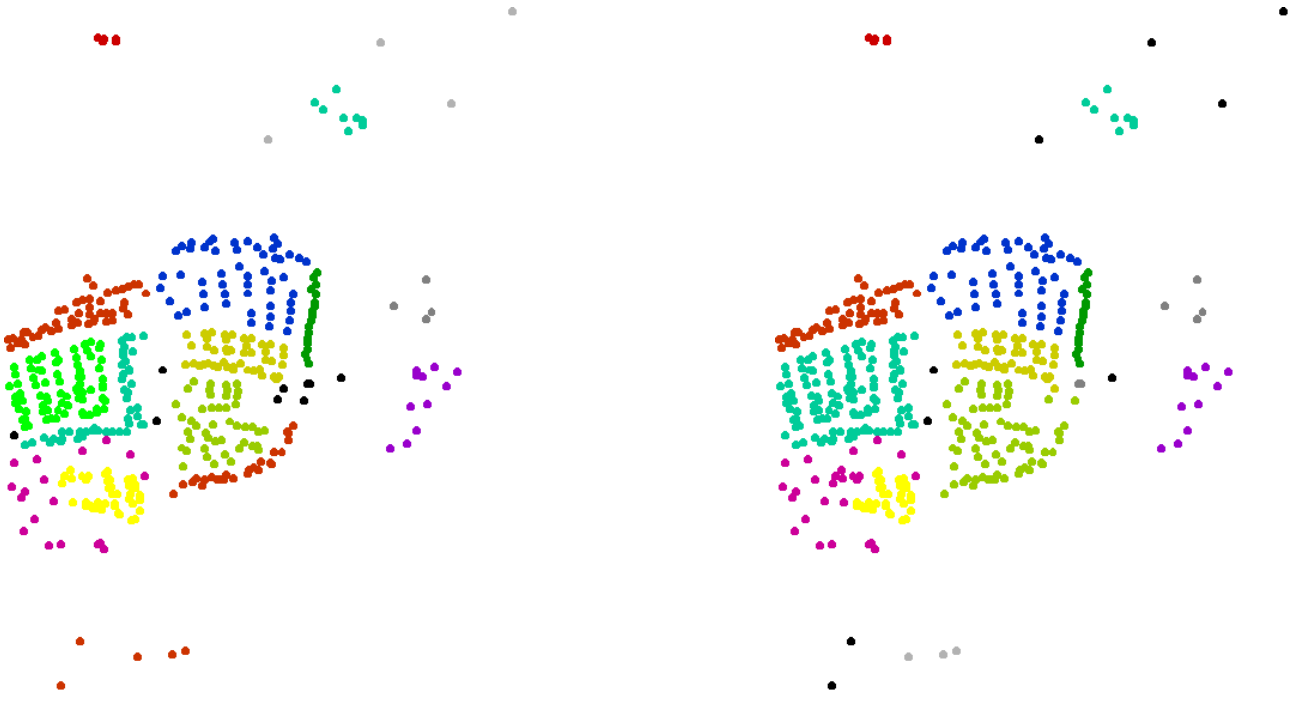
(d) Auswertung 8

Abbildung A.3: Testmuster 1: Auswertungen 5 bis 8



(a) Auswertung 9 (b) Auswertung 10

Abbildung A.4: Testmuster 1: Auswertungen 9 und 10



(a) Auswertung 1 (b) Auswertung 2

Abbildung A.5: Testmuster 2: Auswertungen 1 bis 2

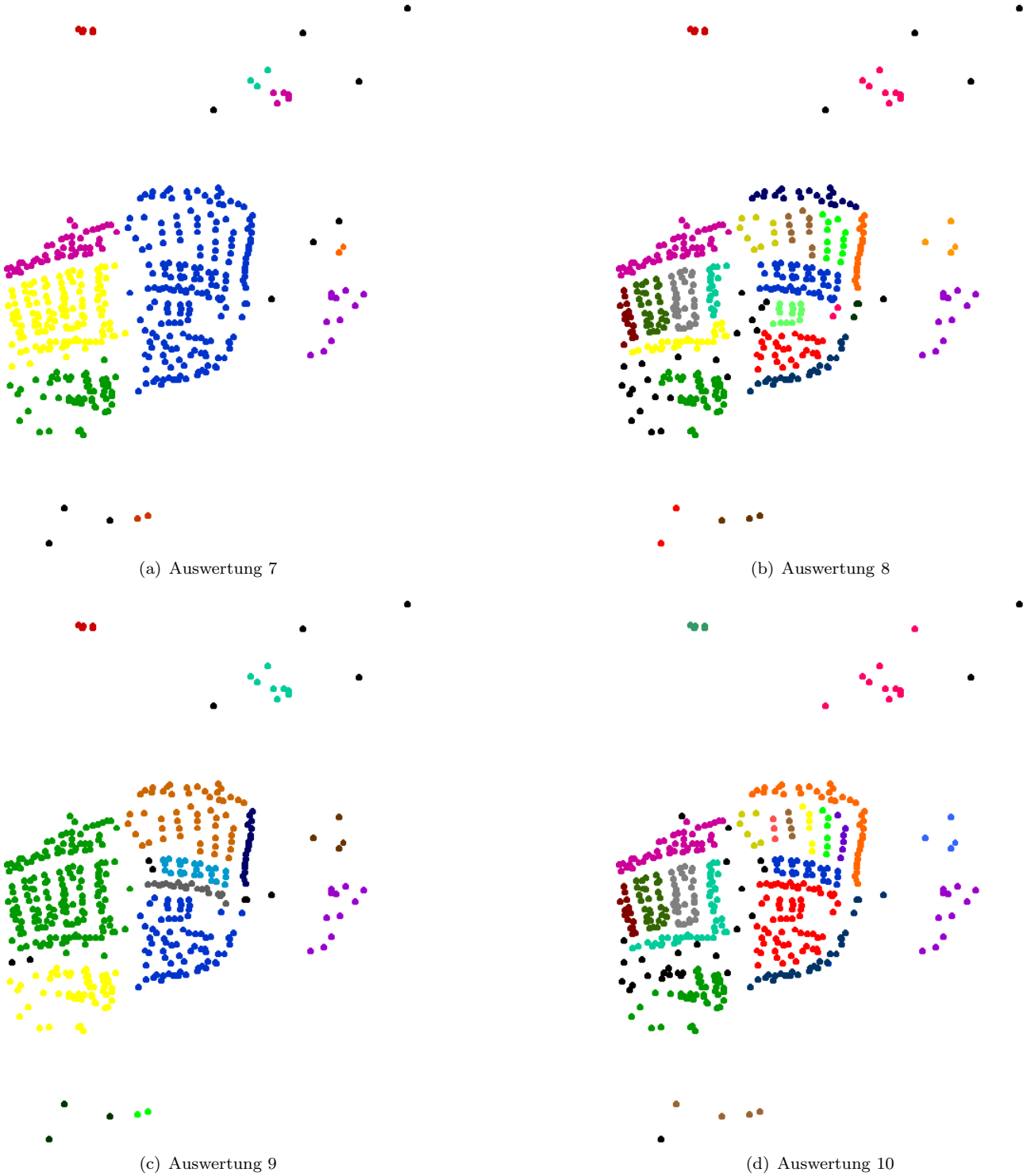
(a) Auswertung 3

(b) Auswertung 4

(c) Auswertung 5

(d) Auswertung 6

Abbildung A.6: Testmuster 2: Auswertungen 3 bis 6



(a) Auswertung 7

(b) Auswertung 8

(c) Auswertung 9

(d) Auswertung 10

Abbildung A.7: Testmuster 2: Auswertungen 7 bis 10

Anhang B

Testmessungen

Die hier aufgelisteten Werte wurden auf einem PC mit einem 1.6 GHz Pentium4 Prozessor und 512 MB Hauptspeicher erzielt. Die Spalten der Tabelle enthalten die folgenden Informationen:

Testdatensatz: Bezeichnung des Testdatensatzes.

#Objekte: Anzahl der Objekte im Testdatensatz.

Modus: Algorithmusvariante, wie in Kapitel 7 beschrieben.

#Cluster: Anzahl der gefundenen Cluster.

#PQ-Loops: Schleifendurchläufe der Prioritätswarteschlange (siehe Kapitel 7).

#Level: Anzahl der durchgeführten Wiederholungen des HPGCL-Algorithmus bis die Endgruppierung gefunden wurde.

Zeit: Gemessene Laufzeit des HPGCL-Algorithmus (ohne Berechnung der Nachbarschaftsgraphen).

Tabelle B.1: Ergebnisse für alle Testdatensätze

Testdatensatz	#Objekte	Modus	#Cluster	#PQ-Loops	#Level	Zeit
Künstliche Testdaten ohne Rauschen	3476	1	12	5197	0	0min:46s:93ms
		2	7	5322	3	0min:47s:686ms
		3	13	5401	0	0min:49s:594ms
		4	9	5548	3	0min:49s:703ms
		5	12	24644	0	0min:50s:515ms
		6	7	24729	3	0min:49s:969ms
		7	13	24729	0	0min:46s:686ms
		8	9	24876	3	0min:48s:594ms
Künstliche Testdaten mit Rauschen	3944	1	280	96293	0	0min:46s:969ms
		2	187	104028	3	0min:52s:30ms
		3	358	94781	0	0min:24s:484ms
		4	328	102762	3	0min:26s:202ms
		5	278	115992	0	0min:54s:656ms
		6	195	123460	3	0min:54s:765ms

Fortsetzung auf nächster Seite

Tabelle B.1 – Fortsetzung von vorheriger Seite

Testdatensatz	#Objekte	Modus	#Cluster	#PQ-Loops	#Level	Zeit
		7	358	100008	0	0min:30s:16ms
		8	323	108298	3	0min:31s:343ms
h01	40	1	12	290	0	0min:0s:15ms
		2	10	393	2	0min:0s:31ms
		3	15	281	0	0min:0s:15ms
		4	12	410	2	0min:0s:31ms
		5	12	268	0	0min:0s:15ms
		6	10	371	2	0min:0s:31ms
		7	14	256	0	0min:0s:15ms
		8	12	382	2	0min:0s:47ms
h02	100	1	28	1507	0	0min:0s:31ms
		2	16	1822	3	0min:0s:94ms
		3	38	1674	0	0min:0s:47ms
		4	32	2232	3	0min:0s:172ms
		5	25	1498	0	0min:0s:31ms
		6	18	1808	3	0min:0s:94ms
		7	37	1637	0	0min:0s:47ms
		8	31	2170	3	0min:0s:156ms
h03	144	1	17	3009	0	0min:0s:62ms
		2	9	3176	3	0min:0s:140ms
		3	39	3238	0	0min:0s:79ms
		4	28	3793	3	0min:0s:265ms
		5	16	3001	0	0min:0s:63ms
		6	13	3137	2	0min:0s:125ms
		7	39	3169	0	0min:0s:78ms
		8	28	3723	3	0min:0s:266ms
h04	474	1	93	31535	0	0min:0s:313ms
		2	37	32808	3	0min:0s:703ms
		3	155	32122	0	0min:0s:359ms
		4	123	34800	3	0min:1s:30ms
		5	92	31598	0	0min:0s:328ms
		6	37	32727	3	0min:0s:718ms
		7	153	32408	0	0min:0s:359ms
		8	127	35116	3	0min:1s:62ms
h05	752	1	155	82450	0	0min:0s:749ms
		2	63	85903	4	0min:1s:624ms
		3	230	80851	0	0min:0s:844ms
		4	187	87501	4	0min:2s:250ms
		5	144	78654	0	0min:0s:812ms
		6	70	81585	4	0min:1s:593ms
		7	228	76705	0	0min:0s:829ms
		8	188	83015	4	0min:2s:250ms
		1	142	51001	0	0min:0s:469ms

Fortsetzung auf nächster Seite

Tabelle B.1 – Fortsetzung von vorheriger Seite

Testdatensatz	#Objekte	Modus	#Cluster	#PQ-Loops	#Level	Zeit
h06	612	2	70	54182	4	0min:1s:140ms
		3	210	48448	0	0min:0s:547ms
		4	172	53313	3	0min:1s:437ms
		5	137	59278	0	0min:0s:500ms
		6	69	53038	4	0min:1s:140ms
		7	203	47650	0	0min:0s:531ms
		8	173	52163	3	0min:1s:407ms
h07	747	1	159	80755	0	0min:0s:703ms
		2	63	85088	4	0min:1s:562ms
		3	264	77045	0	0min:0s:796ms
		4	220	82912	3	0min:1s:875ms
		5	154	79282	0	0min:0s:765ms
		6	74	83285	4	0min:1s:531ms
		7	255	74794	0	0min:0s:749ms
		8	209	80528	3	0min:1s:796ms
h08	294	1	51	12212	0	0min:0s:156ms
		2	19	12896	4	0min:0s:374ms
		3	85	12721	0	0min:0s:171ms
		4	72	14126	3	0min:0s:515ms
		5	51	12256	0	0min:0s:172ms
		6	22	12965	4	0min:0s:422ms
		7	89	12649	0	0min:0s:172ms
		8	72	14115	3	0min:0s:547ms
h09	586	1	127	45916	0	0min:0s:453ms
		2	49	48346	4	0min:1s:78ms
		3	192	46244	0	0min:0s:469ms
		4	159	50239	3	0min:1s:328ms
		5	127	45859	0	0min:0s:469ms
		6	48	47925	3	0min:0s:968ms
		7	182	45892	0	0min:0s:516ms
		8	148	49988	3	0min:1s:250ms
h1-9	3144	1	635	1385004	0	0min:11s:438ms
		2	231	1431813	5	0min:17s:891ms
		3	1032	1371074	0	0min:12s:968ms
		4	839	1441011	4	0min:20s:437ms
		5	608	1363388	0	0min:12s:343ms
		6	261	1402945	5	0min:17s:436ms
		7	997	1341066	0	0min:12s:750ms
		8	824	1401461	4	0min:19s:952ms
3D-Laserdaten	2714	1	239	1327947	0	0min:17s:172ms
		2	73	1333238	4	0min:29s:47ms
		3	899	1279999	0	0min:17s:202ms
		4	738	1321161	3	0min:42s:420ms

Fortsetzung auf nächster Seite

Tabelle B.1 – Abschluss von vorheriger Seite

Testdatensatz	#Objekte	Modus	#Cluster	#PQ-Loops	#Level	Zeit
		5	213	1357761	0	0min:19s:390ms
		6	76	1362221	3	0min:29s:313ms
		7	851	1287560	0	0min:19s:312ms
		8	738	1320229	3	0min:42s:640ms
Testbild	16384	1	70	547767	0	5min:43s:857ms
		2	28	548586	3	20min:26s:786ms
		3	138	543386	0	7min:42s:278ms
		4	110	545902	3	37min:56s:188ms
		5	61	2722200	0	7min:17s:325ms
		6	28	2722843	2	17min:16s:337ms
		7	137	2716345	0	6min:54s:810ms
		8	119	2718313	2	27min:19s:4ms
Luftbild	16728	1	1630	16728	0	10min:46s:465ms
		2	629	38077757	4	13min:16s:822ms
		3	5295	38146380	0	11min:25s:760ms
		4	4411	39178531	5	18min:26s:86ms
		5	1483	39880707	0	13min:59s:901ms
		6	641	40024785	4	16min:4s:572ms
		7	5098	39891107	0	13min:47s:744ms
		8	4415	40692391	5	20min:47s:866ms

Anhang C

Nachbarschaftsgraphen

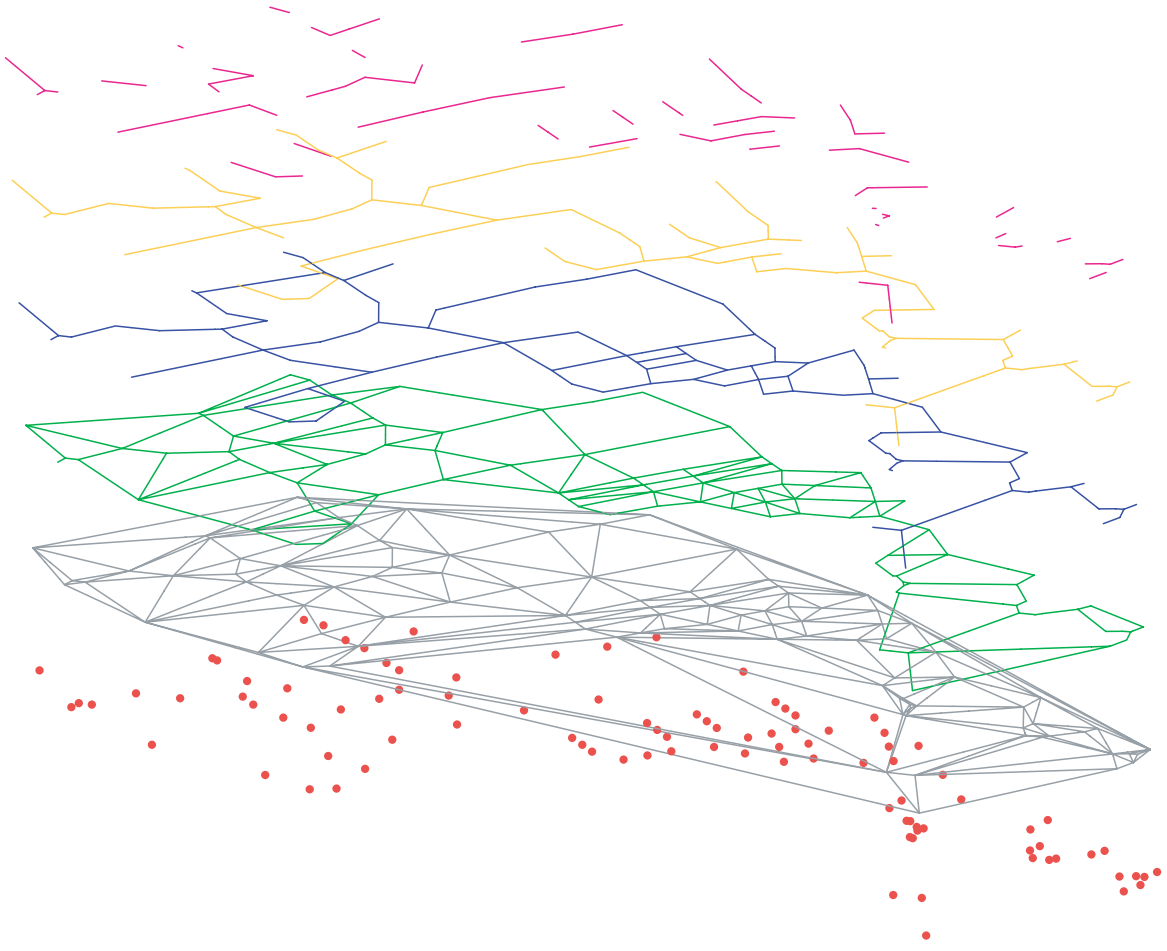


Abbildung C.1: Hierarchie der Nachbarschaftsgraphen: 3D-Darstellung der Teilmengenbeziehung (Von oben nach unten: NNG, MST, RNG, GG, DT, Punktmenge).

(a) Punktmenge

(b) Nächster-Nachbar-Graph

(c) 2-Nächster-Nachbar-Graph

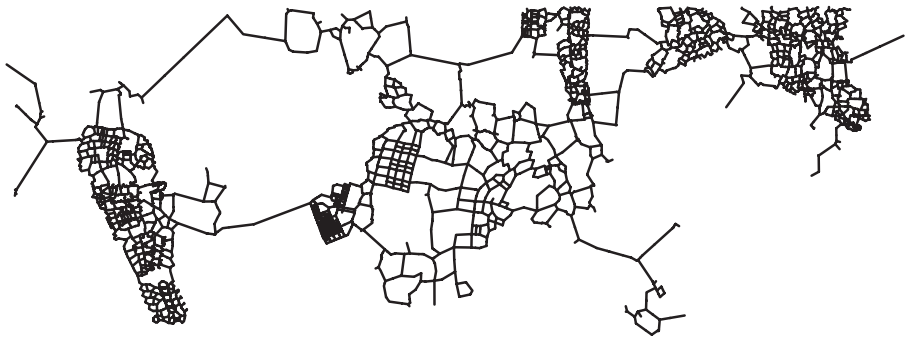
(d) 3-Nächster-Nachbar-Graph

Abbildung C.2: Verschiedene Nachbarschaftsgraphen zum Testgebiet Vaihingen.

(a) 4-Nächster-Nachbar-Graph

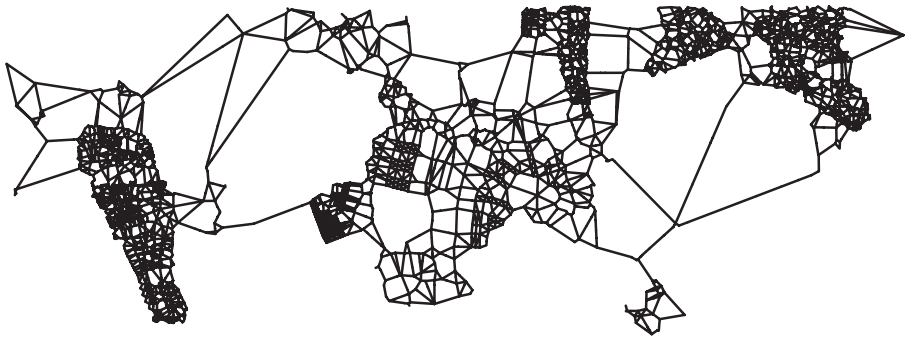
(b) 5-Nächster-Nachbar-Graph

(c) Minimal spannender Baum (MST)

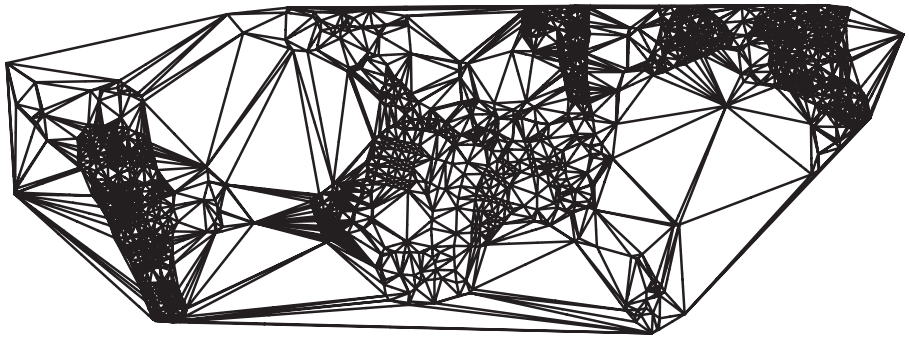


(d) Relativer-Nachbarschafts-Graph (RNG)

Abbildung C.3: Verschiedene Nachbarschaftsgraphen zum Testgebiet Vaihingen.



(a) Gabriel-Graph (GG)



(b) Delaunay-Triangulierung (DT)

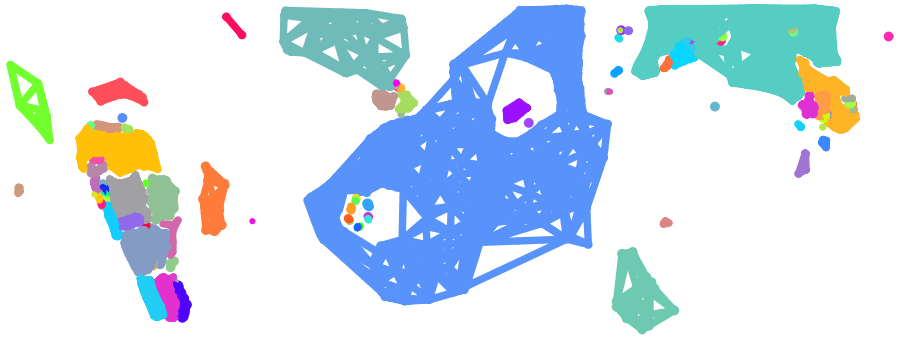
(c) Sphere of Influence Graph (SIG)

Abbildung C.4: Verschiedene Nachbarschaftsgraphen zum Testgebiet Vaihingen.

Anhang D

Auswertung Vaihingen

(a) NNG im Modus 2



(b) MST im Modus 6

Abbildung D.1: Ergebnisse des iterativen HPGCL-Algorithmus.

(a) MST

(b) NNG-MST

(c) RNG

(d) NNG-RNG

Abbildung D.2: Ergebnisse des iterativen HPGCL-Algorithmus für die einzelnen Graphen und Teilhierarchien im Modus 2.

(a) GG

(b) NNG-GG

(c) DT

(d) NNG-DT

Abbildung D.3: Ergebnisse des iterativen HPGCL-Algorithmus für die einzelnen Graphen und Teilhierarchien im Modus 2.

Dank

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter im DFG-Projekt „Semantische Modellierung zur Extraktion topographischer Informationen aus Bildern und Karten“ am Institut für Photogrammetrie der Universität Stuttgart und im „WIPKA-Projekt“ des BKG am Institut für Kartographie und Geoinformatik der Universität Hannover. Der Deutschen Forschungsgemeinschaft sei für die finanzielle Unterstützung meiner Mitarbeit im DFG-Projekt gedankt und dem Bundesamt für Kartographie und Geodäsie danke ich für die finanzielle Unterstützung meiner Mitarbeit im WIPKA-Projekt.

Mein ganz besonderer Dank gilt Herrn Prof. Dieter Fritsch, der diese Arbeit ins Leben gerufen hat und trotz meiner zweijährigen „Industriepause“ immer daran geglaubt hat, dass ich diese Arbeit beenden werde. Frau Prof. Monika Sester möchte ich für ihre konstruktive Betreuung in der Schlussphase dieser Arbeit danken.

Bedanken möchte ich mich auch bei meinen Kolleginnen und Kollegen des Instituts für Photogrammetrie der Universität Stuttgart und des Instituts für Kartographie und Geoinformatik der Universität Hannover für das angenehme Arbeitsklima und die vielen hilfreichen Anmerkungen und Diskussionen. Besonders bedanken möchte ich mich bei denjenigen, die sich für die manuelle Clusterauswertung zur Verfügung gestellt haben und bei denjenigen, die durch aufmerksame Durchsicht und Korrektur bei der Entstehung dieser Arbeit mitgeholfen haben.

Zum Schluss möchte ich mich noch ganz besonders bei meiner Familie für ihre Unterstützung bedanken.

Lebenslauf

Name		Karl-Heinrich Anders
Anschrift		Groß-Buchholzer-Str.14 30655 Hannover
Geburtsdatum/-ort		1.3.1966 in Ludwigsburg
Schulbildung	09/1972 – 07/1976	Grundschule in Ludwigsburg
	09/1976 – 07/1979	Realschule in Marbach
	09/1979 – 07/1982	Realschule in Ludwigsburg
	09/1982 – 07/1985	Technisches Gymnasium in Ludwigsburg
Schulabschluß	05/1985	Abitur
Wehrdienst	10/1985 – 12/1986	
Studium	WS/1986 – SS/1993	Studium der Informatik an der Universität Stuttgart
Studienabschluß	07/1993	Diplom-Informatiker
Beruflicher Werdegang		
	11/1993 – 12/1999	Wissenschaftlicher Mitarbeiter am Institut für Photogrammetrie der Universität Stuttgart
	01/2000 – 03/2002	Software Ingenieur bei der Z/I Imaging GmbH in Oberkochen
	seit 04/2002	Wissenschaftlicher Mitarbeiter am Institut für Kartographie und Geoinformatik der Universität Hannover