Automatische Interpretation von Semantik aus digitalen Karten im World Wide Web

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

Vorgelegt von
Fen Luo
aus Guangxi, China

Hauptbrichter :	Prof. Dr.-Ing. Dieter Fritsch
Mitberichter:	Prof. Dr.-Ing Ralf Bill
Tag der mündlichen Prüfung:	15.12.2014

Institut für Photogrammetrie
der Universität Stuttgart
2014

Inhaltsverzeichnis

Zusammenfassung

Im Internet befindet sich eine sehr große Menge an raumbezogenen Daten, die in Form von Raster- und Vektorkarten unterschiedliche Ausschnitte der Welt darstellen. Die in diesen Karten enthaltenen Informationen sind jedoch nicht automatisch auffindbar, da sie mittels bestimmter Kartenelemente kodiert sind. Ihre Semantik wird erst bei der Interpretation durch einen Betrachter explizit. Die Karteninformationen sollen jedoch nicht nur von Menschen, sondern auch von Maschinen interpretiert werden können. Dies erfordert schon die große Menge der zu interpretierenden Daten. Die automatische Ableitung der Semantik aus den Karten wird unter dem Begriff A*utomatische Karteninterpretation* zusammengefasst. Es handelt sich dabei also um einen Prozess, der implizites Wissen eines Kartenbestandes explizit macht. Hierzu soll die vorliegende Arbeit Lösungen in Form der Karteninterpretation anbieten.

Die Karteninterpretation dieser Arbeit erfolgt an Vektorkarten, die im Internet zu finden sind. Für die gezielte Suche der Vektorkarten des Internets wird eigens ein Webcrawler entwickelt. Der Webcrawler ist eine Suchmaschine, die speziell nach Vektorkarten sucht. Dazu wird ausschließlich das Shapefile-Dateiformat gesucht, das sich zu einer Art Standardformat im GIS-Umfeld entwickelt hat und in dem die Vektorkarten zumeist abgespeichert sind. Um möglichst viele Shapefiles zu finden, wird die Suche auf Servern betrieben, auf denen die Wahrscheinlichkeit Shapefiles zu finden hoch ist. Diese Server werden zuvor durch Google-Suche nach dem Schlüsselwort „shapefile download“ gefunden.

Die Karteninterpretation umfasst Verfahren zur Interpretation der Kartenobjekte, der Kartentypen sowie des Maßstabs.

Zunächst soll das Verfahren zur Interpretation der Objekte einer Karte vorgestellt werden. Hier geht es darum, die Objekte anhand ihrer spezifischen Charakteristika automatisch zu erkennen. Die Objekterkennung basiert auf SOM (Self-Organizing Map), bekannt aus der künstlichen Intelligenz. Die Kartenobjekte werden in Klassen wie beispielsweise Gebäudegrundriss oder Straßennetz gegliedert. Für jede Klasse sollen die ihr jeweils eigenen Merkmale gefunden und in eine der SOM zugängliche Form, hier als Parametervektor, gebracht werden. Die Parametervektoren bilden die Eingabemuster, die in der Lernphase von SOM gelernt werden. Nachdem die Eingabemuster aller Objektklassen von SOM gelernt wurden, wird der Parametervektor für jedes auf der Karte vorliegende Objekt ausgewertet und in die SOM eingegeben. Durch das zunächst erfolgte Lernen der Eingabemuster können die Objekte anhand ihrer jeweils berechneten Parametervektoren der entsprechenden Objektklasse zugeordnet werden.

Als weiteres Verfahren soll die Interpretation des Kartentyps vorgestellt werden. Karten sind nach ihrem inhaltlichen Gehalt und Zweck in Kartentypen wie beispielsweise Flusskarten, Straßenkarten, Höhenlinienkarten etc. kategorisiert. Wie bei der Interpretation der Objekte wird auch hierzu die SOM

verwandt. Es werden also auch Eingabemuster gelernt, die die geometrischen Merkmale der Kartentypen repräsentieren. Die Merkmale ergeben sich sowohl aus der Struktur der einzelnen Objekte als auch aus der Topologie zwischen den Objekten auf einer Karte. Wird nun eine Karte in die SOM eingegeben, so erkennt die SOM anhand des gelernten Eingabemusters den entsprechenden Kartentyp. Zusätzlich erhält man den Dateinamen der Karten sowie den Inhalt der Webseite, auf welcher die Karte gefunden wurde. So wird in der vorliegenden Arbeit ebenfalls untersucht, inwiefern diese Zusatzinformationen bei der Interpretation des Kartentyps helfen können.

Die automatische Interpretation des Maßstabs ist neben der Interpretation der Kartenobjekte und Kartentypen ein weiteres Verfahren, das in der vorliegenden Arbeit diskutiert werden soll. Die Interpretation des Maßstabs wird auf zwei Wegen vorangetrieben: Die Mehrfachrepräsentation und die Detaillierungsgrade. Im ersten Fall kann der Maßstab aus der entsprechenden Repräsentation hergeleitet werden, da ein identisches Objekt in unterschiedlichen realitätsgetreuen Repräsentationen auf der Karte dargestellt wird. Im zweiten Fall kann der Maßstab aus den Detaillierungsgraden abgeleitet werden. Dies basiert darauf, dass die Karten mit verschiedenen Maßstäben unterschiedlich detailliert dargestellt werden.

Abstract

On the Internet there are innumerable spatial data representing different sections of the world in form of raster and vector maps. The information contained in these maps is not automatically discoverable, since it is encoded by means of certain map elements. Its semantics is not explicit unless interpreted by an observer. However, the map information can be interpreted by not only humans but also machines. This already requires the large amount of data to be interpreted. We are going to summarize the automatic derivation of semantics from the maps in terms of *automatic map interpretation*. It involves a process of making the implicit information of a map inventory explicit. For this purpose we present the map interpretation as solutions.

The map interpretation of the current study is done with vector maps what can be found on the internet. For the targeted search of vector maps of the internet, a web crawler is specially developed. The web crawler is a search engine that specifically looks for vector maps. For this, exclusively the shapefile format is sought, which has become a standard format in the GIS environment and in which the vector maps are usually stored. In order to find shapefiles as many as possible, the search is carried out on servers where the probability of finding shapefiles is high. These servers were previously found through the keyword “shapefile download” by Google search.

The maps interpretation includes methods of interpretation of the map objects, of the map types, and of the map scale. First, we will introduce the method of interpreting the map objects. Our aim is to automatically detect the objects based on their specific characteristics. The object recognition is based on self-organizing map (SOM) that is borrowed from artificial intelligence. The map objects are classified into, for example, building floor plan and road network. Its own characteristics should be found for each class and brought in one of the accessible forms of SOM, in this case, a parameter vector. The parameter vectors form the input patterns that are learned in the training phase of SOM. After the input patterns of all object classes of SOM have been learned, the parameter vector is evaluated for each of the present objects on the map and given to the SOM. By the previously successful learning of the input pattern, the objects can be assigned based on each of their calculated parameter vectors of the corresponding object class.

The interpretation of map type is presented as another method. Maps are categorized into different types according to their substantive content and purpose, such as river maps, road maps, contour maps, etc. As for the interpretation of objects, SOM is used here. Hence the input patterns will also be learned which represent the geometric characteristics of the map types. The characteristics arise from both the structure of individual objects and the topology between objects on a map. Now, with a given map in the SOM, the SOM recognizes the appropriate map type according to the learned input pattern. In addition, one obtains the filenames of the maps as well as the content of the website where the map

was found. In the present thesis we also investigated how this additional information can help in the interpretation of map type.

The automatic interpretation of the map scale is a further method in addition to the interpretation of the map objects and map types, which is discussed in the present thesis. The interpretation of the map scale is implemented in two ways: the multi-representation and the details grade. In the former case, the scale of the relevant representation can be derived, where an identical object in different realistic representations on the map is shown; while in the latter case, the scale is derived from the details grade, on the basis of the fact that maps with different scales are displayed on different levels of details.

1 Einführung

Die Nachfrage nach Darstellungsformen raumbezogener Daten ist in den letzten Jahrzehnten rasant angestiegen. Neben den herkömmlichen auf Papier gedruckten Karten existieren neue Möglichkeiten der Visualisierung der Geodaten. Karten können mithilfe neuer Medien, wie beispielsweise dem Computer oder mobilen Endgeräten dargestellt werden. Dazu werden die räumlichen Informationen digital in Datenbanken gespeichert. Mit der deutlich gestiegenen Nutzung des Internets finden sich dort auch immer mehr digitale Karten. Für eine automatische Deutung dieser raumbezogenen Daten sind Analyseinstrumente erforderlich. Ein solches Instrument ist die automatische Interpretation der digitalen Karten.

Bei einer automatischen Karteninterpretation geht es um die automatische Erkennung der geographischen Elemente und ihrer Beziehung untereinander. Die für Menschen sichtbaren Informationen sind nicht immer explizit in einem Datenbestand vorhanden. So können Menschen einen Kreisverkehr oder eine Autobahnkreuzung auf einer Karte erkennen, jedoch ist diese Information nicht explizit in der Datenbank gespeichert. Die automatische Karteninterpretation soll die implizit gespeicherten Informationen explizit machen. Sie findet Anwendungen in Themengebieten wie Kartengeneralisierung, Datenvisualisierung, Stadt- und Raumplanung, Katastrophenschutz, Architektur, etc..

Die Interpretation digitaler Karten ist in Raster- und Vektorkarten zu untergliedern. Rasterkarten werden als Bitmaps bezeichnet und sind mit ihrem bildhaften Informationsgehalt mit der klassischen Papierkarte vergleichbar. Sie bestehen aus Pixeln, denen ein Farb- oder Grauwert zugewiesen wird. Bei der Interpretation der Rasterkarten werden die impliziten Informationen nicht aus den einzelnen Objekten oder Formen, sondern aus den Pixeln abgeleitet. Vektordaten hingegen werden anhand ihrer geometrischen Eigenschaften definiert. Sie werden durch Objekte, wie Punkt, Linie oder Polygon dargestellt. Bei der Interpretation der Vektordaten können sowohl die Geometrie der einzelnen strukturierten Objekte als auch die Beziehungen zwischen diesen Objekten berücksichtigt werden.

1.1 Motivation

Im Internet befindet sich eine unüberschaubare Menge an Geodaten. Eine effektive Suchmaschine zur Suche dieser Karten gibt es bisher nicht. Die existierenden Suchmaschinen wie Google arbeiten im Prinzip mit der Suche in Texten. Der Nutzer muss selbst die Ergebnisse herausfiltern, die relevant sind. Er muss die Karten ihrer Darstellung nach visuell interpretieren, um die Semantik explizit zu erkennen und die gewünschten Informationen zu extrahieren. Bei der riesigen Menge an Daten ist das Durchblättern der Suchergebnisse sehr aufwendig. Um dem Nutzer die Arbeit zu erleichtern, soll eine semantische Suche nach Karteninformationen ermöglicht werden.

Die semantische Suche nach Karteninformationen ist Bestandteil des semantischen Webs. Das Semantische Web wird als die nächste Evolutionsstufe des Internets angesehen [Berners-Lee et al. 2001]. Hierbei werden die im Internet zugänglichen Dokumente mit einer Semantik versehen, so dass die Dokumente bezüglich ihrer Bedeutung durchsucht werden können. Für das semantische Web wird eine Reihe von Techniken entwickelt, um die Inhalte von Websites mit logischen Beziehungen zu belegen. Die zwei wesentlichen Elemente hierbei sind zum einen die Auszeichnungssprachen und zum anderen die Abfragesprachen. Die Auszeichnungssprache kann eine Ontologie anhand einer formalen Beschreibung erstellen und die Ontologie kann mittels der Abfragesprache in beliebigen Richtungen angefragt werden.

Mithilfe der Auszeichnungssprache können die interpretierten Karteninformationen formal beschrieben werden. So ist beispielsweise zu beschreiben, dass eine Karte einen großen Maßstab enthält, oder dass eine Karte eine Flusskarte ist. Die Abfragesprache ermöglicht Suchanfragen, wie beispielsweise „finde alle Flusskarten, die einen großen Maßstab enthalten“. Nun können alle Karten selektiert werden, die entsprechend ausgezeichnet wurden. In der vorliegenden Arbeit wird die semantische Suche nicht untersucht. Die Arbeit zielt auf die Interpretation der impliziten Informationen aus Karten. Diese Informationen können als Basis für eine semantische Suche dienen.

1.2 Aufgabenstellung

Ziel der vorliegenden Arbeit ist die Entwicklung von Verfahren und Methoden zur automatischen Interpretation von Vektorkarten. Bei den nun erstmalig entworfenen Verfahren dieser Arbeit sollen Vektorkarten mit dem ESRI-Shapefile Format für die Untersuchung aufbereitet werden. Im Internet findet sich eine sehr große Menge an Shapefiles, die als Datenquelle der Arbeit verwendet werden können. Die Suche nach bestimmten Formaten im Internet ist nur eingeschränkt möglich. Google kann z.B. nach den Formaten: pdf, ps, dwf, kml, kmz, xls, ppt, rtf und swf, jedoch nicht nach Esri-Shapefiles suchen. Bei den Formaten .kml und .kmz handelt es sich zwar um Geodaten, dabei liegt der Fokus aber nur auf den Koordinaten der wichtigen Standpunkte. Daher sind diese nicht für die Karteninterpretation geeignet. Bisher existiert keine Suchmaschine mit der Suchmöglichkeit nach Shapefiles auf dem Markt. Die vorliegende Arbeit hat nun ein Webcrawler entwickelt, der diese Aufgabe erfüllt.

Der Schwerpunkt der Interpretation der Kartenobjekte und Kartentypen liegt in der Ermittlung der geometrischen Merkmale. Bei der Charakterisierung für die Kartenobjekte spielen sowohl die quantitativen Merkmale wie absolute Größe der Fläche oder Umfang, als auch die qualitativen Merkmale wie rechter Winkel oder Netzförmigkeit eine Rolle. Für die Kartentypen haben die quantitativen Merkmale keine Bedeutung, da die absoluten Größen unterschiedlicher Karten aufgrund verschiedener Maßstäbe sehr verschieden sein können. Weiterhin werden die topologischen Beziehungen betrachtet. Beispielsweise sind die Knotentypen eine wesentliche Eigenschaft für die Unterscheidung der verschiedenen linienförmigen Kartentypen.

Nachdem Merkmale extrahiert wurden, besteht die Aufgabe darin, anhand der Merkmale die Kartenobjekte bzw. Kartentypen mittels einer Self-Organizing Map (SOM) zu erkennen. Die SOM lernt zunächst an Mustern so lange, bis sie von sich aus in der Lage ist, unbekannte Objekte zu erkennen. Das „Lernen“ der Merkmale wird in der Trainingsphase realisiert. Das „Erkennen“ folgt in der Ausführungsphase. Dabei sollen passende Netzparameter wie beispielsweise die Netzgröße oder Lernschritte festgelegt werden, um die SOM richtig zu konfigurieren. Des Weiteren ist es sinnvoll zu untersuchen, inwiefern sich die Auswahl der Merkmale auf die Interpretation auswirkt.

Nach der Interpretation der Kartenobjekte und Kartentypen wird der Frage nachgegangen, wie sich der Maßstab automatisch erkennen lässt. Der Maßstab soll mittels der Mehrfachrepräsentation und der Detaillierungsgrade abgeleitet werden. Hiermit besteht die Aufgabe darin, die mit dem Maßstab zusammenhängenden Mehrfachrepräsentationen sowie die Detaillierungsgrade zu ermitteln und ihren Zusammenhang aufzudecken.

1.3 Aufbau der Arbeit

In Kapitel 2 wird auf die Grundlagen der verwendeten Technologien eingegangen. Dabei handelt es sich um den Webcrawler für die Suche nach Karten im Internet und um die künstlichen neuronalen Netze zur Erkennung der Kartenobjekte und Kartentypen.

In Kapitel 3 wird zunächst der Stand der Forschung im Bereich der Karteninterpretation aufgearbeitet. Da das Spatial Data Mining für die Karteninterpretation angewendet werden kann, werden wichtige Arbeiten des Themengebiets vorgestellt. Des Weiteren wird ein Überblick über den Stand der Forschung der verwendeten Technologien Webcrawler und künstliche neuronale Netze gegeben.

In Kapitel 4 geht es um die Suche der Vektorkarten im Internet mittels eines Webcrawlers. Zu Beginn wird die Interpretation der Rasterkarten evaluiert. Die Limitierung, die die Interpretation der Rasterkarten mit sich bringt, wird aufgezeigt. Aus dieser Limitierung ergibt sich für die vorliegende Arbeit die Interpretation der Vektorkarten.

Kapitel 5 stellt ein auf den künstlichen neuronalen Netzen basierendes Verfahren für die Interpretation der Kartenobjekte vor. In dem Kapitel wird diskutiert, welche Merkmale der Kartenobjekte zur Interpretation geeignet sind und wie eine SOM zu konfigurieren ist. Am Beispiel der Testkarten werden Ergebnisse präsentiert und eine Bewertung vorgenommen.

Eine ähnliche Gliederung besitzt Kapitel 6, in dem das Interpretationsverfahren für Kartentypen verwendet wird. Für Kartentypen mit linienförmigen sowie polygonförmigen Objekten wird jeweils die Auswahl der Parameter vorgestellt. Schließlich werden die aus den Testkarten gewonnenen Ergebnisse diskutiert.

Kapitel 7 zeigt Ansätze zur Interpretation des Kartenmaßstabs auf. Dabei handelt es sich um die maßstababhängigen Mehrfachrepräsentationen sowie die Detaillierungsgrade. Die Interpretation des Maßstabs aus der Mehrfachrepräsentation wird anhand des Beispiels „Kreisverkehr“ erläutert, während aus den Detaillierungsgraden zwei Sichtweisen, einmal auf die Anzahl der Stützpunkte, das andere Mal auf den Abstand zwischen den Linien erläutert werden.

Abschließend wird in Kapitel 8 eine Zusammenfassung der vorliegenden Forschungsarbeit und ein Ausblick für weitere Untersuchungsmöglichkeiten gegeben.

2 Grundlagen

In diesem Kapitel wird auf die zwei theoretischen Grundlagen der Karteninterpretation eingegangen. Zum einen geht es um die Technologie Webcrawler für die Internetsuche. Zum anderen wird das für die Interpretation eingesetzte Verfahren der künstlichen neuronalen Netze erläutert.

2.1 Webcrawler

Der Webcrawler wird zur Suche nach Vektorkarten im World Wide Web eingesetzt. Ein Webcrawler ist ein Programm, das automatisch Webseiten aus dem World Wide Web durchsuchen und analysieren kann. Es ist ein wichtiger Bestandteil der Suchmaschinen. Der Webcrawler fungiert zuoberst als Informationsquelle der Suchmaschine. Die Webseiten werden zuerst durch die Webcrawler aus dem Internet abgefangen, dann präsentieren die Suchmaschinen wie z.B. Google nach einer aufwendigen Reinigung und Sortierung dem Nutzer die Suchergebnisse.

2.1.1 Grundlagen

Ein Webcrawler legt in der Regel eine bestimmte oder mehrere Webseiten als Startseite fest. An dieser Stelle fängt der Webcrawler an zu arbeiten. Der Webcrawler extrahiert die Links auf der Startseite und durchsucht die durch die Links gezeigten Webseiten. Die Links auf diesen Webseiten werden wiederum extrahiert, damit der Webcrawler weitere Webseiten durchsuchen kann. Dieser Arbeitsprozess wiederholt sich ständig, bis die Voreinstellung zum Beenden getroffen wird.

Die Abbildung 2-1 zeigt den Grundprozess eines Webcrawlers. Das Frontier wird zuerst mit einer Menge von Start-URLs (Seeds) initialisiert. Ausgehend von dieser Startseite sollen alle in ihr befindlichen URLs in dem Frontier gespeichert werden. Die URLs werden an den Downloader überreicht und dieser lädt dann die Webseiten zu den URLs aus dem Internet herunter. Danach übergibt der Downloader die Webseite an den Parser, damit der Parser die vorhandenen URLs auf der Webseite extrahieren kann. Die durch den Parser gefundenen URLs werden schließlich zum Frontier hinzugefügt. Dieser Prozess wird solange iteriert, bis sich keine URL mehr im Frontier befindet. Die Begriffe Frontier, Downloader und Parser werden im Folgenden näher erläutert.

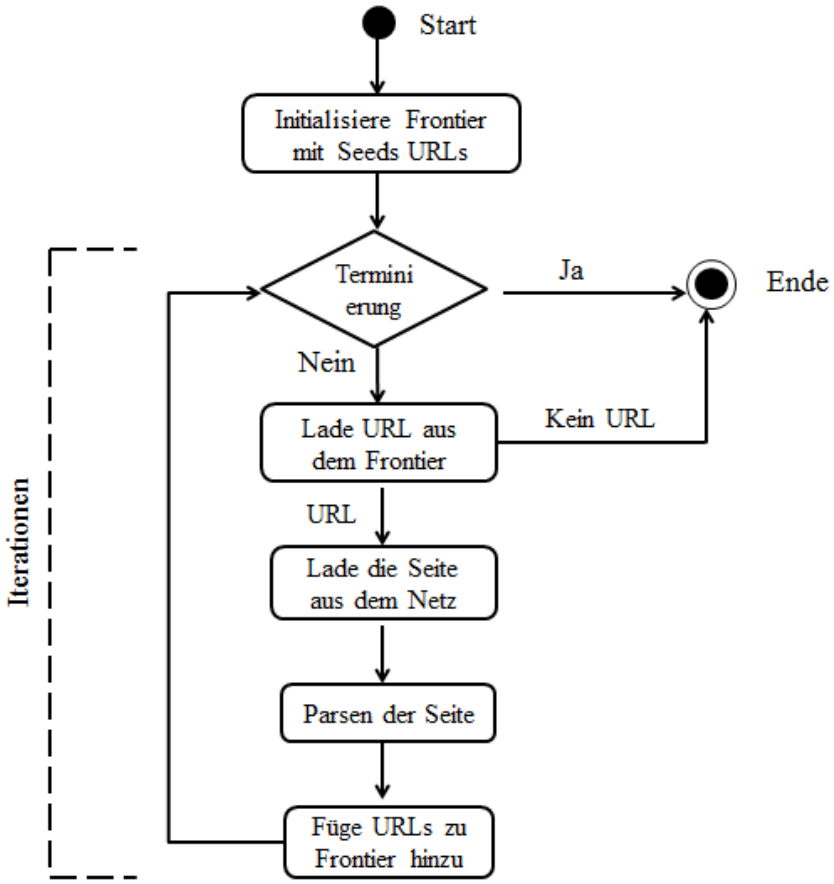


Abbildung 2-1: Ablauf des Webcrawlers (nach [Pant et al. 2004])

2.1.1.1 Frontier

Das Frontier dient dem Webcrawler als eine Datenstruktur, in welcher die nicht besuchten URLs enthalten sind. Die URLs werden in dem Frontier gespeichert und nach einer bestimmten Strategie für den nächsten Schritt in der Crawling-Schleife weitergereicht. Ein Frontier kann als FIFO-Frontier (First-In-First-Out) oder als Priority-Queue implementiert werden.

1. ***FIFO-Frontier***

Bei einem FIFO werden die URLs, die zuerst hinzugefügt werden, auch zuerst aus dem Frontier entnommen. Diese Datenstruktur wird auch als Warteschlange bezeichnet. Es gibt drei wesentliche Strategien, nämlich: Breitensuche (breadth-first), Tiefensuche (depth-first) und eine Kombination von beiden (best-first). Die Implementierung des FIFOs kann mit der Breitensuche und der Tiefensuche aus der Graphentheorie (siehe Anhang A) realisiert werden.

Breitensuche

Bei der Breitensuche geht der Webcrawler von der Startseite aus und analysiert alle URLs auf dieser Ebene, anstatt in eine Ebene tiefer zu gehen. Nachdem alle URLs auf der Ebene bearbeitet werden, wählt der Webcrawler erst eine URL aus und analysiert alle URLs auf der von der URL verlinkten Seite. Die Breitensuche ist einfach zu realisieren, hat jedoch den Nachteil, dass viele irrelevante Webseiten dabei heruntergeladen werden.

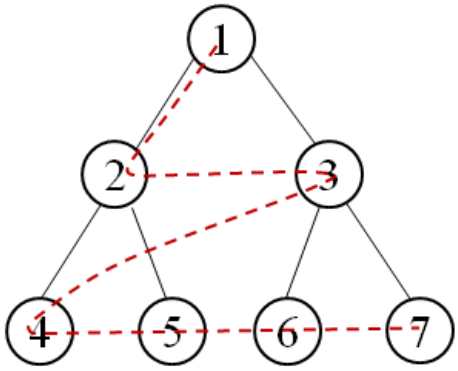


Abbildung 2-2: Breitensuche

Tiefensuche

Bei der Tiefensuche geht der Webcrawler von der Startseite aus und wählt eine URL aus. Die URLs in der von dieser URL verlinkten Seite werden analysiert und eine davon wird ausgewählt. Der Webcrawler folgt dieser und analysiert dessen URLs. Dieser Vorgang wiederholt sich solange, bis die ganze Kette von URLs in der Tiefe durchgearbeitet ist. Danach wird eine neue Kette bearbeitet. Die Gestaltung der Tiefensuche ist relativ einfach, allerdings gibt es ein Problem: Die Links der oberen Stufe sind meist am wertvollsten, je tiefer die Links in der Kette sind, desto mehr sinkt ihr Wert.

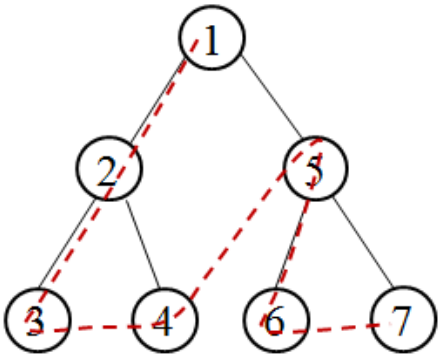


Abbildung 2-3: Tiefensuche

2. ***Priority-Queue***

Eine Vorrangwarteschlange ist eine erweiterte Warteschlange. Die URLs können mit Schlüssel bzw. Priorität versehen werden, um die Reihenfolge der Bearbeitung zu bestimmen. Wenn die Seiten der Priorität nach besucht werden, dann wird die Strategie als die Kombination der Breiten- und Tiefensuche bezeichnet.

Bei der Kombination der Breiten- und Tiefensuche werden URLs nach Ähnlichkeit mit der Ziel-URL oder der Relevanz des vorher definierten Themas verglichen. Der Vergleich der Ähnlichkeit stammt aus der Theorie vom Vector Space Model [Salton & McGill 1983]. Das Thema der Suche und die heruntergeladenen Webseiten werden als Vektoren präsentiert. Der Webcrawler berechnet den Zusammenhang zwischen den heruntergeladenen Webseiten und Stichwörtern des Themas, um die Priorität der URL auf der Webseite zu extrahieren. Somit werden die URLs in der Warteschlange sortiert. Je mehr eine heruntergeladene Webseite mit dem Thema in Verbindung steht, desto enger beziehen sich die URLs aus der Webseite auf das Thema. Der Zusammenhang kann durch das Kosinus-Ähnlichkeitsmaß berechnet werden:

$$\cos(T, W) = \frac{T \cdot W}{|T| \cdot |W|}$$

T: Vektor des Themas

W: Vektor der Webseite

Der Wert -1 steht für einen schwachen, der Wert 1 für einen starken Zusammenhang. Eine solche Strategie kann die Anzahl der irrelevanten Seiten stark reduzieren.

2.1.1.2 Downloader

Die URLs werden von dem Frontier an den Downloader weitergegeben und heruntergeladen. Um eine Webseite herunterzuladen, wird ein Auftrag an den in der URL angegebenen Server gesendet. Die Antwort wird dann von dem Server zurückgeschickt. Für das Senden an den Server und die Antwort aus dem Server kann eine s.g. Timeout-Einstellung festgelegt werden. Dieser Parameter definiert die maximale Zeit der Herstellung der Verbindung, um die Verarbeitungsgeschwindigkeit des Webcrawlers zu erhöhen. Der Wert soll jedoch nicht zu klein eingestellt werden, so dass nicht zu wenig Verbindungen herstellt werden.

2.1.1.3 Parser

Nachdem eine Webseite heruntergeladen wurde, wird sie an den Parser überreicht. Hier werden die URLs auf der Webseite untersucht und extrahiert. Ist dies geschehen, werden die URLs an den Frontier weitergegeben und somit beginnt der Webcrawler den nächsten Arbeitsschritt.

2.1.2 Website und Webcrawler

Mit Webcrawlern können zwar Informationen schnell gefunden werden, jedoch wird durch die Verarbeitung der Anfragen sowie die Auslieferung der Ergebnisse beim Server eine Last erzeugt. Während ein Webcrawler eine Website besucht, wird ein Teil der Leistung der Website geschwächt, so dass die anderen Nutzer bzw. Programme die Website nicht effizient besuchen können. Daraufhin wird das Robots-Exclusion-Standard-Protokoll von einer unabhängigen Gruppierung entwickelt.

Mit dem Robots-Exclusion-Standard-Protokoll kann ein Website-Betreiber festlegen, welche Webseiten in der Domain von einem Webcrawler indexiert werden dürfen und welche nicht. Die Protokolle werden in der Datei robots.txt beschrieben. Die Datei befindet sich normalerweise unter dem Stammverzeichnis. Beispielsweise ist die Datei robots.txt für die Website der Universität Stuttgart unter „http://www.uni-stuttgart.de/robots.txt“ zu finden. Falls die Datei nicht existiert, dürfen alle Webseiten in einer Domain aufgefunden werden.

Die Datei ist eine nach einem bestimmten Schema aufgebaute Textdatei. In der ersten Zeile wird die ID des Webcrawlers unter dem Eintrag „User-agent“ angegeben.

Jeder Webcrawler hat eine eigene Identifizierung, die beim Besuchen einer Website mitgeteilt wird. So lautet die Identifizierung der Google-Suchmaschine z.B. „GoogleBot“, die der Baidu-Suchmaschne „BaiDuSpider“ und die der Yahoo-Suchmaschine „Inktomi Slurp“. Der Website-Manager kann aus dem User Access Logging (UAL) lesen, welcher Webcrawler wann da war und welche Daten abgefragt wurden. Falls ein Webcrawler mit unzulässigem Besuch entdeckt wird, kann der Webcrawler anhand seiner ID geblockt werden.

Der „User-agent“ Eintrag folgt Anweisungen, die für diesen Webcrawler gelten. Mit dem „Disallow“ Eintrag werden die Dateien sowie Verzeichnisse festgelegt, die nicht aufgefunden werden sollen.

Die meisten Websites wollen dass sie von Suchmaschinen gefunden werden können, so dass mehr Nutzer durch das Auffinden mittels der Suchmaschinen die Website besuchen. Damit die Websites vollständiger indexiert werden können, kann eine Karte über die Website, die s.g. Site Map, erstellt werden. In der Site Map können alle URLs der Website aufgelistet werden. Viele Webcrawler nehmen die Site Map als Eingang zu einer Website. So können die Webcrawler praktisch alle Webseiten in der

Website indexieren. Ein Übersehen von Webseiten kann dadurch vermieden und die Last auf Servern kann gemindert werden.

2.1.3 Aktualisieren der Webseiten

Da die Inhalte einer Website sich stets ändern, wird ein Webcrawler aufgefordert, die Änderung der gefangenen Webseiten zu aktualisieren. Der Webcrawler soll nach einem bestimmten Zyklus die Website besuchen, um zu prüfen, welche Webseiten geändert, welche hinzugefügt wurden und welche nicht mehr existieren.

Die Festlegung des Updatezyklus kann die Vollständigkeit der Indexierung stark beeinflussen. Wenn der Zyklus zu lang gesetzt wird, können die neu entstandenen Webseiten nicht aufgefunden werden. Wenn der Zyklus zu kurz ist, wird der Server belastet. Der Zyklus kann für unterschiedliche Websites variieren. Für wichtige Websites kann der Zyklus kurz z.B. ein paar Stunden sein, während er für irrelevante Websites mehrere Monate betragen kann.

Im Normalfall muss der Webcrawler bei der Aktualisierung die Webseiten nicht erneut anfangen. Für die meisten Webseiten ist es ausreichend, die Attribute der Webseiten wie z.B. das Datum zu prüfen. Die Attribute werden mit den letzten Attributen verglichen, sind sie identisch, muss keine Aktualisierung stattfinden.

2.2 Künstliche Neuronale Netze

In diesem Kapitel wird in die Theorie der künstlichen neuronalen Netze eingeführt. Die künstlichen neuronalen Netze bestehen aus mehreren idealisierten Neuronen. Die Neuronen simulieren die biologischen Neuronen und dienen dazu, die Information aufzunehmen und über gerichtete Verbindungen an die anderen Neuronen weiterzuleiten. Die Lernfähigkeit ist ein wesentliches Element der künstlichen neuronalen Netze. Sie können selbständig aus Trainingsbeispielen lernen, ohne dass das neuronale Netz dazu explizit programmiert werden muss [Zell 1997]. Die künstlichen neuronalen Netze finden ihre Anwendung häufig im Bereich Funktionsapproximation, Klassifizierung, Bildverarbeitung und Mustererkennung.

2.2.1 Grundlagen

Das Vorbild der künstlichen neuronalen Netze sind die Nervenzellen aus der Biologie. Die Informationen werden von Milliarden Neuronen im Gehirn verarbeitet und gespeichert. Die Neuronen analysieren die elektrischen Eingangsreize über die Dendriten im Zellkörper, wo die gewichteten Eingangsreize angebracht werden (Abbildung 2-4). Die Neuronen geben einen elektrischen Impuls über das Axon weiter, wenn die Summe der Eingangsreize einen bestimmten Grenzwert überschreitet.

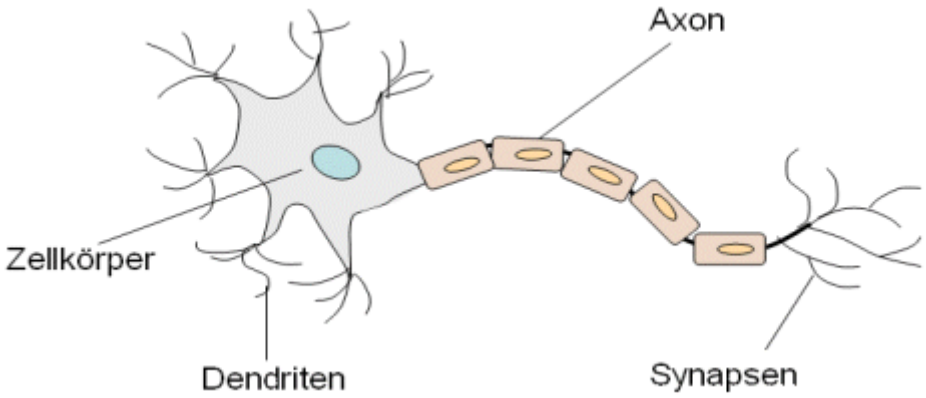


Abbildung 2-4: Biologisches Neuron (nach [Krause 1993])

Die Technologie der künstlichen neuronalen Netze hängt jedoch wenig mit dem biologischen Vorbild zusammen, da vieles für die neuronalen Netze vereinfacht wird. Die künstlichen Neuronen empfangen über Verbindungen die Informationen anderer Neuronen. Analog zum biologischen Neuron wird das künstliche Neuron in den künstlichen neuronalen Netzen verwandt (Abbildung 2-5). Aus den Eingangssignalen x_i wird eine gewichtete Summe gebildet, aus der mithilfe der Aktivierungsfunktion f der Ausgang berechnet wird.

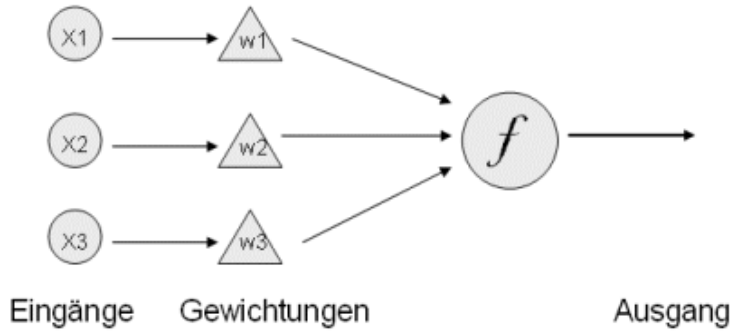


Abbildung 2-5: Künstliches Neuron

Die Eingänge können durch die Berechnung mit den Gewichtungen durch die Aktivierungsfunktion f gehemmt oder erregt werden. Aktivierungsfunktionen sind beispielsweise die Schwellenwertfunktion, die sigmoide Funktion oder die lineare Funktion (siehe Anhang B).

2.2.2 Lernen

Die künstlichen neuronalen Netze zeichnen sich durch ihre Lernfähigkeit aus. In einem Lernprozess geht es um die Informationsverarbeitung, um durch das Lernen der Trainingsmuster die Gesetzmäßigkeit der Eingabedaten herauszufinden. Ein wichtiger Teil in einem Lernprozess ist die Bestimmung der Gewichtsveränderungen zwischen den Neuronen. Die Gewichtung bezeichnet die Stärke der Ver-

bindung zwischen den Neuronen. Die Anpassung der Verbindungsgewichtung nach einem Lernverfahren ermöglicht die Verarbeitung der unbekannten und nicht trainierten Eingabedaten. Unterschiedliche Lernregeln definieren unterschiedliche Gewichtsveränderungen. Im Folgenden werden zwei Lernalgorithmen vorgestellt.

Überwachtes Lernen

Beim überwachten Lernen gibt ein „externer Lehrer" zu jedem Eingabemuster in der Trainingsmenge ein bekanntes, korrespondierendes Ausgabemuster vor. Der „Lehrer" überwacht und steuert gleichzeitig den Lernzustand der neuronalen Netze [Hertz et al. 1991]. Das Ziel des überwachten Lernens ist der Abgleich von berechneter Ausgabe und richtiger Ausgabe. Die ermittelten Abweichungen werden an die künstlichen neuronalen Netze übermittelt und während des Trainings verkleinert. Beispiele sind die Delta-Lernregel und die Backpropagation (siehe Anhang C).

Unüberwachtes Lernen

Beim unüberwachten Lernen gibt es keinen „Lehrer". Die Trainingsmenge enthält keine paarweise korrespondierenden Ein- und Ausgabemuster. Das Netz versucht selbstständig die Eigenschaften der Eingabedaten zu extrahieren und die Daten zu klassifizieren. Die neuronalen Netze werden nicht darüber informiert, ob die Eingabemuster richtig verarbeitet werden. Das Lernen findet hierbei völlig ohne Kontrolle statt. Typische Beispiele sind die Hebb-Lernregel (siehe Anhang C) und die SOM (siehe 2.2.4).

2.2.3 Netztopologien

Die künstlichen neuronalen Netze werden aus vielen miteinander verbundenen künstlichen Neuronen aufgebaut. Beim Aufbau stehen unterschiedliche Netztopologien zur Verfügung. Dabei unterscheiden sich die Art und Anzahl der Einheiten sowie deren Verbindungen. Künstliche neuronale Netze können nach der Netzwerkstruktur und Verbindungsausrichtung in Feedforward-Netz und Rekurrentes-Netz aufgeteilt werden:

Feedforward-Netz

Das Feedforward-Netz ist das Netz ohne Rückkopplungen. Es arbeitet ausschließlich von der Eingabeschicht in Richtung zur Ausgabeschicht. Zwischen der Eingabeschicht und Ausgabeschicht könnten mehrere verdeckte Schichten existieren. Eine Schicht ist immer mit der nächst höheren Schicht verbunden. Die SOM ist ein Beispiel für das Feedforward-Netz. Das Feedforward-Netz kann in einschichtiges und mehrschichtiges Netz aufgeteilt werden:

1. *Einschichtiges feedforward-Netz:* Das Netz besitzt lediglich eine Ausgabeschicht und ist damit die einfachste Struktur der künstlichen neuronalen Netze (Abbildung 2-6).

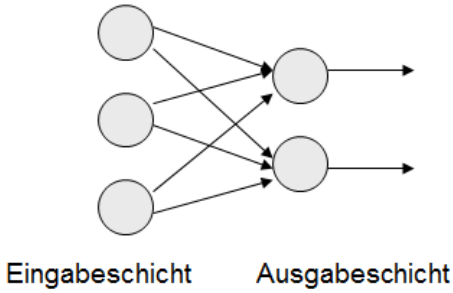


Abbildung 2-6: Einschichtiges Feedforward-Netz

2. *Mehrschichtiges feedforward-Netz*: Das Netz besteht aus einer Eingabeschicht, einer oder mehrerer versteckter Schichten und einer Ausgabeschicht.

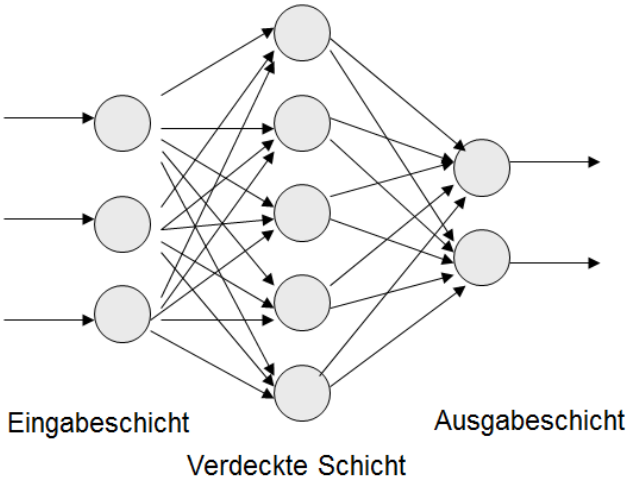


Abbildung 2-7: Mehrschichtiges Feedforward-Netz

Die Abbildung 2-7 zeigt ein Beispiel der Netztopologie. In der verdeckten Schicht sind interne Neuronen enthalten, die die Information innerhalb des Netzes verarbeiten und von außen nicht aufrufbar sind.

Rekurrentes Netz

Im Gegensatz zum Feedforward-Netz erlaubt das Rekurrente-Netz Rückkopplungen zwischen den einzelnen Schichten (Abbildung 2-8).

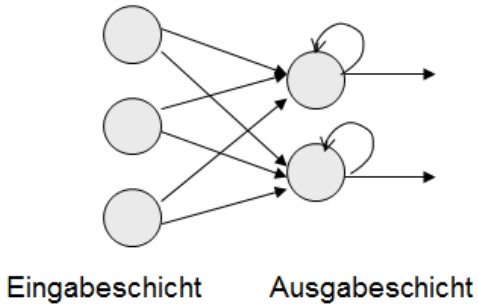


Abbildung 2-8: Rekurrente-Netz

2.2.4 Netztypen

Die neuronalen Netze unterscheiden sich durch verschiedene Netztopologien und Lernregeln. Es gibt jedoch keine klare Zuordnung zwischen Netztyp und Lernregel, da einige Netztypen auf gleiche Lernregel zurückgreifen, während andere unterschiedliche Lernregeln verwenden. Im Anhang D werden die folgenden Netztypen vorgestellt:

- Perzeptron
- Pattern Associator
- Hopfield-Modell

2.2.5 Eigenschaften

Die neuronalen Netze besitzen sowohl positive als auch negative Eigenschaften. Positive Eigenschaften sind beispielsweise:

- Lernfähigkeit
- Parallelität
- Verteilte Wissensrepresentation
- Höhere Fehlertoleranz
- Robustheit gegenüber Störungen
- Teilweise biologische Plausibilität
- Aktive Repräsentation

Negative Eigenschaften sind z.B.:

- Wissenserwerb ausschließlich durch Lernen
- Keine Analyse möglich
- Großer Rechenaufwand

Auf die genannten Eigenschaften wird im Anhang E näher eingegangen.

2.2.6 Selbstorganisierende Karten

Die selbstorganisierende Karte (Self-Organizing Map, SOM) gehört zu der Gruppe der künstlichen neuronalen Netze. Sie wurde in den frühen 1980er Jahren von Prof. Teuvo Kohonen entwickelt und wird oft nach seinem Name als Kohonennetz bezeichnet.

In der Arbeit wird die SOM für die Karteninterpretation verwendet, da die SOM sich durch das unüberwachte Lernen als vorteilhaft kennzeichnet. Außerdem ist die SOM näher am menschlichen Gehirn orientiert. Die Einsicht in die Funktionsweise des menschlichen Gehirns macht sie geeignet für Interpretationsaufgaben, bei denen menschliche Betrachtung simuliert werden soll. Der Einsatz der SOM im Bereich Data Mining, Bildverarbeitung, Datenverarbeitung, Sprachverarbeitung und Robotersteuerung ist seit langem als vielversprechend anzusehen. Die SOM wird in der vorliegenden Arbeit daraufhin untersucht, ob sie brauchbare Ergebnisse in der Karteninterpretation liefern kann.

2.2.6.1 Biologie

Die SOM simuliert die prinzipielle Funktionsweise der menschlichen Großhirnrinde. Die Gebiete der Großhirnrinde werden u.a. nach Sprachzentrum, somatosensorischem Kortex, auditivem Kortex, visuellem Kortex aufgeteilt.

Die Theorie der SOM kann beispielsweise anhand des sensorischen Kortex wiedergegeben werden. Der sensorische Kortex ist zuständig für die Auswertung der sensorischen Eingaben des Körpers. Beispiele dafür sind Berührung, Druck und Temperatur. Die Region für die Sinneswahrnehmungen des Arms ist im somatosensorischen Kortex mit nebeneinander gelegenen Bereichen für Finger, Hand, Unterarm und den Ellenbogen.

Die häufiger genutzten Bereiche (Hand) entsprechen stärkerer Ausprägung als die weniger genutzten Bereiche (Unterarm, Ellenbogen). Die stärkere Ausprägung ermöglicht eine detailliertere Verarbeitung der Wahrnehmungen. Bei einem Eintreffen eines Reizes in einem Bereich wird der Reiz in der entsprechenden Region, in der Hirnrinde von den Neuronen, welche in Nachbarschaftsbeziehungen zu einander liegen, verarbeitet. Die Nachbarschaftsbeziehung zwischen den Neuronen ist ein wesentliches Merkmal der SOM.

2.2.6.2 Prinzip

Mit der SOM können die hoch dimensionalen Daten in wenigen Dimensionen dargestellt werden, d.h. der Raum der Eingabedaten besitzt wesentlich höhere Dimensionen als der Raum der Ausgabedaten. Auch bei der SOM wird, wie bei anderen künstlichen Neuronalen Netzen, zwischen Lernphase und Ausführungsphase unterschieden.

In der Lernphase wird die SOM durch Eingabemuster trainiert und lernt neue Eingabedaten in der Ausführungsphase zu verarbeiten. Die Neuronen der Ausgabeschicht werden durch die Eingabeneuronen, die der Dimension des Eingabevektors entsprechen, gereizt. Jedes Neuron der Ausgabeschicht wird durch die Eingabeneuronen erregt oder gehemmt.

Die Eingabeneuronen werden durch Gewichtung mit den Ausgabeneuronen verbunden. Jedes Neuron der Ausgabeschicht besitzt jeweils eine Gewichtung und die Festlegung der Gewichtung ist je nach der Problemstellung flexibel. Der Eingabevektor eines Eingabeneurons wird mit allen Gewichtsvektoren verglichen. Das Ausgabeneuron mit der höchsten Ähnlichkeit gewinnt und wird als Erregungszentrum (oder „Siegerneuron“) bezeichnet. Um dieses Neuron herum sind ähnliche Neuronen angesiedelt. Dieser Vergleich mit den Gewichtungen stellt sich als Hauptaufgabe in der Lernphase dar. Nach der Lernphase werden in der Ausführungsphase die Ähnlichkeiten anderer Eingabeneuronen zu den Siegerneuronen berechnet. Das Siegerneuron mit der größten Ähnlichkeit wird dem Eingabeneuron zugeordnet. Auf dieser Weise wird das neue Eingabeneuron automatisch klassifiziert.

2.2.6.3 Topologie

Die SOM ist eine einschichtige Karte. Die Eingabeschicht wird durch Neuronen dargestellt, die durch Verbindungsgewichtung mit allen Ausgabeneuronen verbunden sind (siehe Abbildung 2-9).

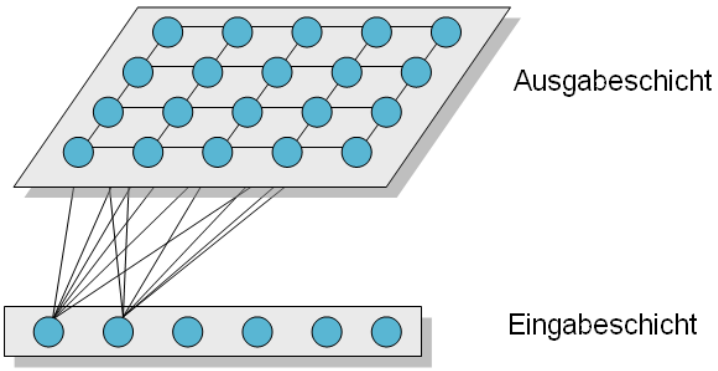


Abbildung 2-9: SOM

Die Neuronen der Ausgabeschicht sind untereinander verbunden und stehen in einer festen Nachbarschaftsbeziehung zueinander. Die Nachbarschaft ist eine wichtige Eigenschaft der SOM. Sie dient dazu, die Nachbarn des Erregungszentrums zu identifizieren. Die Nachbarschaft kann durch einen Radius festgelegt werden. Der Radius ändert sich in Radiusformen von Dreieck, Kreis, Sechseck etc.. Die Abbildung 2-10 zeigt die Einschränkung eines Nachbarschaftsradius in Viereck an. Die Nachbarschaft wird in der Abbildung mit dunkler Hintergrundfarbe dargestellt.

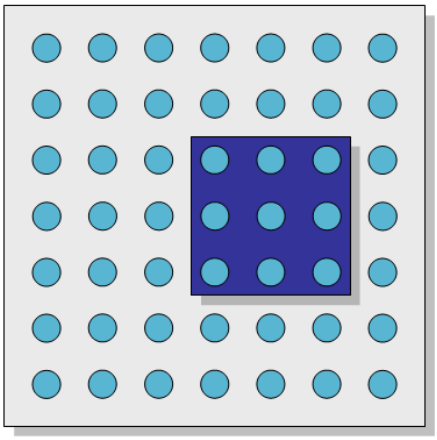


Abbildung 2-10: Nachbarschaftsradius

Für die Berechnung der Nachbarschaft sind die bereits bewährten Nachbarschaftsfunktionen anwendbar. Im Folgenden einige Beispiele dafür:

- Die Zylinderfunktion

$$h_{zylinder}(z,r) = \begin{cases} 1 & wenn\ z < r \\ 0 & sonst \end{cases}$$

z: Abstand zwischen einem Neuron und dem Erregungszentrum

r: Nachbarschaftsradius

- Die Gaußsche Glockenfunktion

$$h_{gauss}(z,r) = e^{-(z/r)^2}$$

- Die Mexican-Hat Funktion. Die Mexican-Hat-Funktion ist die 2. Ableitung der normalisierten Gaussfunktion.

$$h_{gauss2}(z,r) = (1 - (\frac{z}{r})^2)e^{-(z/r)^2}$$

Die Topologie und die Anzahl der Neuronen, die die Genauigkeit der SOM beeinflussen, werden von Anfang an festgelegt. Die Anzahl der Neuronen in der Ausgabeschicht darf nicht kleiner als die Anzahl der zu erkennenden Klassen gewählt werden und wird üblicherweise so gewählt, dass mehrere Neuronen zu einer gleichen Klasse zugeordnet werden.

2.2.6.4 Trainingsphase

Das Training kann in eine Groborientierungs- und eine Feinabstimmungsphase unterteilt werden. In der Groborientierungsphase wird am Anfang die Gewichtung in der Ausgabeschicht mit Zufallswerten initialisiert. Danach wird die Ähnlichkeit für einen Eingabevektor zu jedem Gewichtsvektor von der SOM bewertet und das Siegerneuron sowie seine umliegenden Neuronen werden bestimmt. Für die Bewertung der Ähnlichkeit werden das Cosinus-Ähnlichkeitsmaß und die Euklidische Distanz am häufigsten verwendet. Die Euklidische Distanz wird mit folgender Formel berechnet:

$$d_j = \sqrt{\sum_{i=1}^{n} (x_i - w_{ij})^2}$$

x: Eingabevektor

w: Gewichtung

In der Feinabstimmungsphase werden die Gewichtsvektoren des Siegerneurons sowie die seiner umliegenden Neuronen optimiert. Dadurch wird die Ausgabeschicht in jedem weiteren Lernschritt feiner angepasst. Die Gewichtungen werden nach folgender Formel optimiert:

$$\Delta w_{ij} = \eta * h_{cj} * (\mathrm{x_i} - \mathrm{w_{ij}})$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}$$

η: *Lernrate*

h_{cj}: *Nachbarschaftsfunktion*

Die Nachbarschaft spielt dabei eine Rolle, damit die anfangs erwähnte Topologie erhalten werden kann. Außerdem richtet sich die Funktion nach der Lernrate, die eine zeitliche Funktion bezeichnet. Mit einem großen Radius kann eine Grobstruktur für die SOM am Anfang gebildet werden. Diese Grobstruktur soll jedoch mit Fortschreiten des Lernens durch einen immer kleiner werdenden Radius verfeinert werden. Aus diesem Grund soll die Nachbarschaftsfunktion mit der Lernrate, die im weiteren Verlauf immer kleiner wird, multipliziert werden. Die Lernrate liegt immer im Bereich von null bis eins.

Wenn die Gewichtungen für alle Eingabemuster berechnet werden, beginnt ein neuer Lernschritt, in dem die Gewichtungen erneut berechnet werden. Dieser Vorgang wiederholt sich so lange, bis eine Abbruchbedingung erfüllt ist, z.B. wenn die Lernrate gleich null ist.

2.2.6.5 Qualität

Mehrere Methoden wurden bereits entwickelt, um die Qualität der SOM abzuschätzen. Zwei oftmals verwendete Methoden sind die Quantisierungsfehler (quantization error) und topographischen Fehler (topographic error) [Kohonen 2001].

Der Quantisierungsfehler wird im Allgemein für Vektorquantisierungs- und Clusterverfahren eingesetzt. Bei der Betrachtung des Quantisierungsfehlers werden die Topologie und Anordnung der Karte ignoriert. Der Quantisierungsfehler wird aus den Distanzen zwischen Vektoren ermittelt und misst im Falle der SOM die durchschnittliche Entfernung zwischen dem einzelnen Neuron und seinem am besten passenden Neuron.

$$\varepsilon_q = \frac{1}{N}\sum_{i-1}^{N} \|X_i - m_c\|$$

X: Eingabevektor

c: Indiziert das beste zugeordnete Neuron für X

m: Gewichtung des am besten zugeordneten Neurons

Der Quantisierungsfehler bewertet den Einbau der SOM in die Daten. Der kleinste durchschnittliche Quantisierungsfehler wird für eine optimale Karte erwartet. Je kleiner der Quantisierungsfehler ist, desto kleiner ist der Mittelwert der Entfernung von dem Eingabevektor zum Erregungszentrum.

Um die topologische Erhaltung der SOM zu beurteilen, wird der topographische Fehler untersucht, mit dem die Kontinuität der Zuordnung von der Eingabeschicht für die SOM betrachtet wird.

$$\varepsilon_t = \frac{1}{N}\sum_{k=1}^{N} \mu(X_k)$$

Die Funktion $\mu(X_k)$ ist 1, wenn das beste und das zweitbeste Erregungszentrum für ein Neuron nicht benachbart sind, ansonsten ist sie 0. Je niedriger der topographische Fehler ist, desto besser ist die Topologie der SOM.

3 Stand der Forschung

In diesem Kapitel wird ein Überblick über die einschlägigen Arbeiten in den für die vorliegende Arbeit signifikanten Forschungsgebieten gegeben. Dabei werden Verfahren zur Karteninterpretation und zum Spatial Data Mining vorgestellt. Weiterhin werden Verfahren der verwendeten Technologien Webcrawler und künstliche neuronale Netze präsentiert.

3.1 Karteninterpretation

Die Karteninterpretation kann in Raster-basierte und Vektor-basierte Karten aufgeteilt werden. Die relevanten Arbeiten werden im Folgenden aufgearbeitet.

3.1.1 Rasterkarten

Die Interpretation der Rasterkarten wurde bereits häufig untersucht. Beispielsweise konzentriert sich das Verfahren von [Callier & Saito 2011] auf die Erkennung der Straßen einer Rasterkarte. Angesichts der überlappenden Features einer Rasterkarte ist es problematisch, die Straßen automatisch zu interpretieren. Das Verfahren reduziert im ersten Schritt das Farbrauschen auf der Karte. Als nächstes werden die linearen Features ermittelt, um die Pixel einer möglichen Straße zu prüfen. Diese Pixel werden als initiale Zellen für das Straßennetzwerk verwendet. Zum Schluss werden mehrere Pixel mittels eines farbigen Histogramms extrahiert, um das Ergebnis zu verbessern.

[Kou et al. 2012] ermitteln ein Verfahren, das über die Erkennung der Straßen hinaus die Erkennung der Straßentypen wie Autobahn, Hauptstraße und Schienen etc. ermöglicht. Eine Rasterkarte beinhalte Kartenelemente wie z.B. unterschiedliche Straßentypen, Flüsse, Annotationen wie Ortsnamen und weitere Karteninhalte. Ein Typ des Kartenelements wird auf der Karte in gleicher Farbe dargestellt, demzufolge können die Kartenelemente durch Farbcluster voneinander getrennt werden. Da die getrennten Kartenelemente durch verschiedene Annotationen gedeckt oder getrennt werden, sind diese explizit zu erkennen. Die Annotationen werden in der Arbeit nicht als Bildrauschen bearbeitet, sondern durch die Farbcluster analysiert. Die durch die Annotation unterbrochenen Straßenabschnitte können mithilfe der Straßeneigenschaften wieder verbunden werden.

Die Erkennung einer Straßenkreuzung aus der Rasterkarte wird von [Chiang et al. 2009] untersucht. Hier werden Methoden der Bildverarbeitung und Bilderkennung in die automatische Trennung der Straßenkreuzung von anderen Symbolen oder Linien auf der Rasterkarte eingesetzt. Die Pixel auf dem Vordergrund werden zuerst von der Karte getrennt. Im Anschluss werden die Pixel entfernt, die nicht Bestandteil einer Straße sind, um die Straßen zu extrahieren. Die extrahierten Straßen werden nun mit einem Operator aus der Bildverarbeitung wiederaufgebaut. Durch die Anzahl und Orientierung der Straße an einer möglichen Kreuzung werden die wirklichen Kreuzungen entdeckt.

Die Vektorisierung der Rasterkarte erfolgt ebenfalls in der Arbeit von [Meinel et al. 2006]. Vor der Vektorisierung werden zuerst die Wohngebäude aus der rasterbasierten digitalen Topographischen Karte 1:25.000 selektiert. Diese Gebäudeextraktion wird durch digitale Bildverarbeitung in einem mehrstufigen Prozess vollautomatisch realisiert. Nach der Gebäudeselektion werden diese vektorisiert, um die Gebäude mit geometrischen Attributwerten in vordefinierten Typen zu klassifizieren. Für die regelbasierte Klassifikation werden ca. 20 Kennzahlen, insbesondere Gebäudefläche, -umfang, -länge, -breite, Umkreisfläche, Kompaktheit, Kreis- und Rechteckähnlichkeit, Orientierung und Abstand zum Nachbargebäude berechnet. Anhand der Kennzahlen werden alle Wohngebäude (Einfamilien-, Doppel-, Reihen-, Mehrfamilienhaus geschlossen, Mehrfamilienhaus offen, Zeilen-, Punkt-, Hochhausbebauung, Villen) durch ein Fuzzy-Entscheidungsnetzwerk typisiert (Abbildung 3-1).

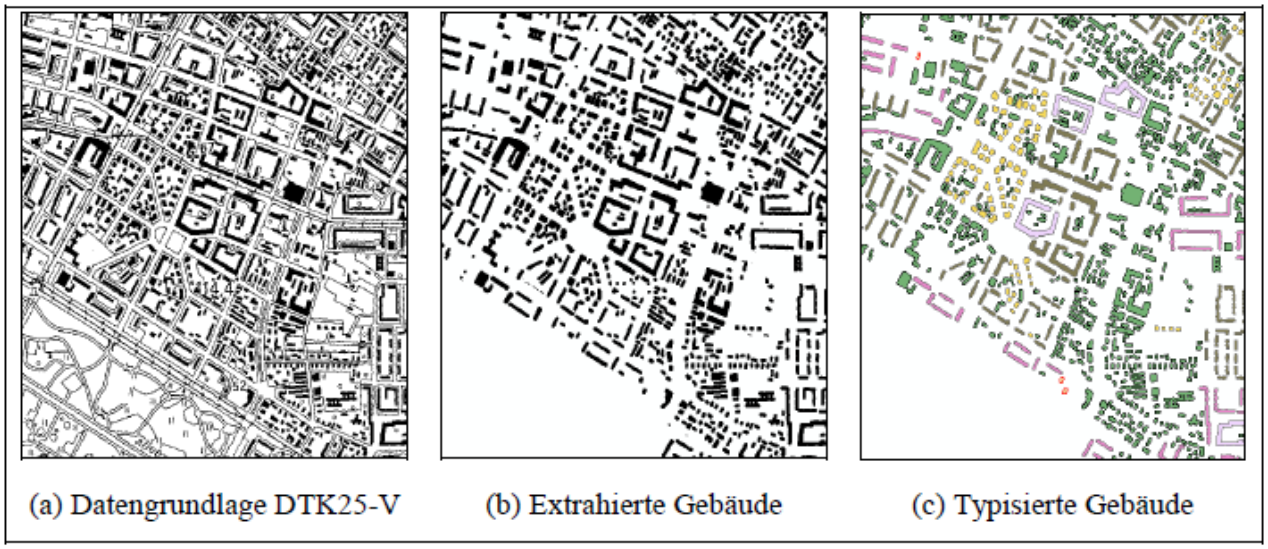


Abbildung 3-1: Automatische Gebäudedetektion, -vektorisierung und -typisierung (Beispiel Dresden-Striesen) (kopiert mit freundlicher Genehmigung aus [Meinel et al. 2006])

Ein weiterer Ansatz der Karteninterpretation auf rasterbasierten Karten wird von [Graeff & Carosio 2002] präsentiert. Mit dem Ansatz wird eine Abfragesprache SQL (Structured Query Language) aus der Datenbankanwendung auf Rasterdaten eingesetzt. Um die Objekte und andere Informationen aus Rasterdaten direkt abfragen zu können, wird ein durch Abfragen steuerbares Mustererkennungssystem entwickelt. Das Mustererkennungssystem wird durch Template Matching, Fuzzy-Logik und andere Ansätze in ihrer Kombination realisiert.

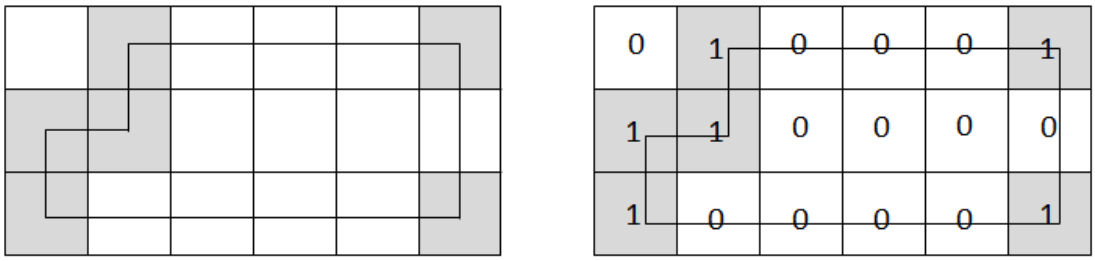
Die Mustererkennung findet in der Karteninterpretation häufig ihren Einsatz. Beispielsweise wird in der Arbeit von [Herold et al. 2012] ein erweiterter Algorithmus der Mustererkennung eingesetzt, um eine objektbasierte Bildanalyse durchzuführen und die zeitliche Änderung der Karte zu erkennen. Die Arbeit von [Frischknecht et al. 1998] untersucht die automatische Interpretation der topographischen Karten im Rasterformat ebenfalls auf der Basis der Mustererkennung.

Weitere Verfahren der Interpretation der Rasterkarten befinden sich z.B. in der Arbeit von [Khotanzad & Zink 2003]. Dort werden die Höhenlinien und andere geographische Informationen extrahiert.

[Cao & Tan 2002] entwickeln eine Methode basierend auf der Theorie, dass Text aus kürzeren Abschnitten als die Grafik besteht, um Text von Grafik auf einer Karte zu trennen. [Hai & Bao 2008] erkennen Straßen, indem Pixel auf einer Rasterkarte in drei Gruppen: Straße, nicht-Straße und Bildrauschen getrennt werden. Zum Bildrauschen gehören Texte, Symbole etc.. [Dhar & Chanda] teilen die Karte in verschiedene Schichten auf und erkennt die symbolischen Darstellungen anhand ihrer spezifischen geometrischen und morphologischen Eigenschaften auf jeder Schicht. In [Viglino & Pierrot-Deseilligny 2003] werden die Katasterkarten von der RGE – eine Datenbank vom IGN (French National Geographic Institute) interpretiert.

3.1.2 Vektorkarten

Auch bei den Vektorkarten sind bereits mehrere Ansätze zur Interpretation untersucht worden. Ein Beispiel ist in [Lannon et al. 2007] zu finden. Der Algorithmus erkennt den Typ und das Alter der Gebäude auf einer digitalen Karte für das Model EEP (Energy and Enviroment Prediction) [Jones et al. 2000]. Für die Erkennung der Gebäudegrundrisse wird die Form eines Gebäudetyps zunächst in der Orientierung und Längsrichtung normalisiert. Danach wird die Ähnlichkeit eines Gebäudes mit der normalisierten Form verglichen, dabei ist eine mögliche Spiegelung und Verzerrung erlaubt. Für das Vergleichen wird ein Gitter über den Umriss des Gebäudes gelegt. Die Zellen, in denen die Gebäudeecken sich befinden, werden als 1 markiert. Die anderen Zellen werden als 0 markiert (Abbildung 3-2). In diesem Fall wird der Umriss des Gebäudes in den Code 010001110000100001 umgewandelt.



a) Umriss des Gebäudes mit Gitter b) Code der Zellen

Abbildung 3-2: Typischer Umriss des Gebäudes (nach [Lannon et al. 2007])

In der Arbeit von [Lüscher et al. 2008] wird die Ontologie zur Erkennung der räumlichen Muster und Strukturen auf städtischer Fläche angewendet. Die Ontologie wird definiert, indem die Muster wie z.B. der Gebäudetyp anhand der Attribute wie Fläche, Ausrichtung etc. textuell beschrieben werden. Die Ontologien können wiederum zur Erkennung solcher Muster in den Vektorkarten genutzt werden.

[Steinhauer et al. 2001] präsentieren die automatische Interpretation der abstrakten Regionen auf einer Karte durch hierarchische Beschreibungen. Eine abstrakte Region besteht aus mehreren Karten-objekten, die in einem einzelnen Objekt gruppiert werden können. Der Prozess wird in zwei Schritte aufgeteilt. Im ersten Schritt werden die Kandidaten der Region mit einer einfachen Regel, nämlich die der parametrisierten Nachbarschaft extrahiert. Im zweiten Schritt werden die Kandidaten mit anderen Beziehungen getestet.

Die vorliegende Arbeit wurde in ihrem Ansatz erstmals in [Walter & Luo 2011] veröffentlicht. In der Veröffentlichung wird die automatische Interpretation digitaler Karte diskutiert und ein weiteres Verfahren [Walter 2007] erläutert. Das Verfahren basiert auf einem Grid-basierten Clustering-Ansatz. Die Identifikation der Cluster erfolgt auf einer städtischen Vektorkarte und wird als Ergebnis auf eine Rasterkarte dargestellt. Am Anfang wird ein Operator mit zwei Parameter definiert: Die Größe der Zellen für die resultierende Rasterkarte und der Radius des Clusters. Dann wird die Fläche in gleichmäßige und quadratische Zellen aufgeteilt. Um die städtische Fläche zu identifizieren, werden die Knoten-Dichte und die Rechtwinkligkeit der Straßen als Merkmale ausgewählt. Es wird davon ausgegangen, dass das städtische Gebiet mehrere topologische Knoten und weniger orthogonale Straßen als das ländliche Gebiet besitzt. Ein Indikator berechnet die Knoten-Dichte und die Rechtwinkligkeit für jede Zelle und stellt das Ergebnis durch Grauwert auf der resultierenden Rasterkarte dar. Die Untersuchung mit der Knoten-Dichte führt zu einem sehr guten Ergebnis - die mit der Rechtwinkligkeit zu einem weniger guten Ergebnis. Dies liegt daran, dass die Annahme, Städte enthalten mehrere irreguläre und unorthogonale Straßen, nicht immer zutrifft. Zur Optimierung der Ergebnisse können jedoch mehrere Merkmale nach Gewichtung kombiniert und die Parameter des Operators variiert werden.

In [Heinzle et al. 2007] wird ein Ansatz für die automatische Erkennung der Muster in Straßennetzwerken auf städtischen Gebieten präsentiert. Die Liniengeometrien repräsentieren die Straßenverbindungen und werden mithilfe graphengestützter Ansätze untersucht. Als Schwerpunkt werden strich-, gitter-, stern- und ringförmige Strukturen (Abbildung 3-3) in der Arbeit analysiert und beschrieben. Für die automatische Erkennung der Strukturen werden Algorithmen entwickelt und in einem wxWidget-tool integriert.

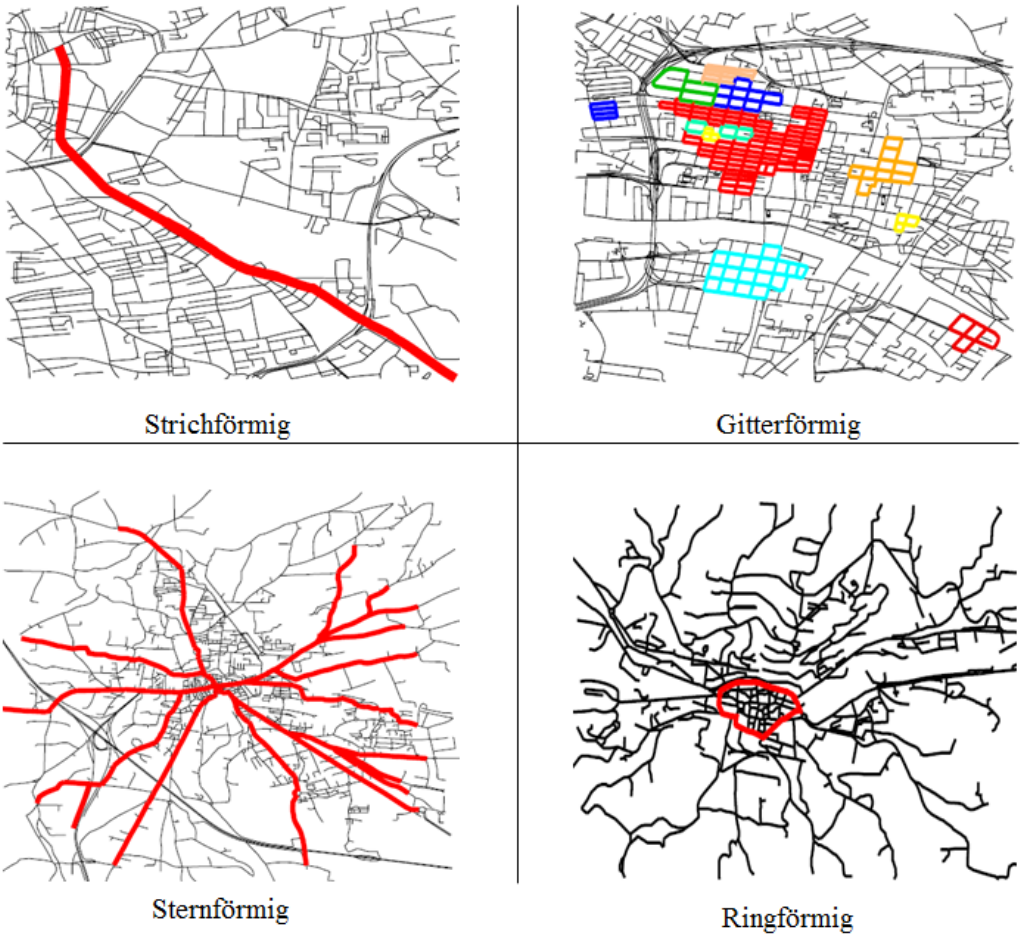


Abbildung 3-3: Straßenstruktur (kopiert mit freundlicher Genehmigung aus [Heinzle et al. 2007])

Ein auf geometrischer und logischer Beschreibung basierendes Verfahren zur automatischen Interpretation wird in der Arbeit von [Zhang et al. 2008] untersucht. In dieser Arbeit wird das high-level Wissen der topographischen Datenbestände in maschinen-lesbares Low-Level-Wissen formalisiert. Um das Wissen flexibel zu interpretieren und die bereits entwickelten Funktionalitäten wieder verwenden zu können, wird das Wissen in fünf Typen [Ruas & Lagrange 1995] eingeteilt: Geometrisches Wissen; topologisches Wissen; semantisches Wissen; prozessuales Wissen und strukturelles Wissen. Die logische Beschreibung der Karte ist auch die Basis des Verfahrens aus [Lanza et al. 2002]. Die logische Beschreibung der Kartenmuster wird durch rechnerische Methoden generiert und dann von einem Lernverfahren verwendet.

3.2 Spatial Data Mining

Unter Spatial Data Mining versteht man die systematische Anwendung von Methoden auf einen Datenbestand mit dem Ziel der Gewinnung impliziter und vorher unbekannter Information. Die Methoden sind meist statisch-systematisch begründet und finden die Objekte anhand der Mustererkennung aus der räumlichen Datenbank. Durch eine Sprache werden die Muster wie z.B. Klasse, Assoziieren,

Regel und Cluster beschrieben. Dabei wird die morphologische und räumliche Beziehung zwischen den Objekten berücksichtigt.

Das Spatial Data Mining ist eine Zentralkomponente der raumbezogenen Knowledge Discovery. Da Methoden der Spatial Data Mining auf die Karteninterpretation eingesetzt werden können, werden die Techniken des Spatial Data Minings für die vorliegende Arbeit studiert.

Beim Spatial Data Mining können verschiedene Techniken eingesetzt werden. In dieser Arbeit werden zwei wichtige Techniken und zwar Clustering und Klassifikation näher beschrieben. Clustering ist die Aufgabe der Gruppierung der Objekte einer Datenbank in vorher nicht bestimmte Klassen (Cluster), während die Klassifikation die Aufgabe der Einordnung der Objekte in vordefinierte Klassen ist.

3.2.1 Clustering

Clusterverfahren werden häufig im Spatial Data Mining verwendet. Das Ziel der Clusterverfahren besteht darin, die Objekte entsprechenden Clustern zuzuordnen, so dass eine möglichst große Ähnlichkeit in einem Cluster und eine möglichst geringe Ähnlichkeit zwischen Clustern bestehen. Cluster-Verfahren können in verschiedenen Kategorien eingeteilt werden:

- *Hierarchische Verfahren*: Sie basieren auf der Berechnung der Distanz. Cluster bestehen aus Objekten, die zueinander eine kleinere Distanz haben als zu den Objekten anderer Cluster. Hierarchische Verfahren können in top-down (aufteilendes, divisives) Clustering und bottom-up (aufhäufendes, agglomeratives) Clustering aufgeteilt werden. Top-down Clustering beginnt mit allen Objekten in einem Cluster und spaltet rekursiv ein Cluster in zwei Cluster auf bis eine bestimmte Clusteranzahl erreicht ist. Bottom-up Clustering beginnt mit jedem Punkt als ein Cluster und verschmelzt rekursiv zwei Cluster zu einem größeren Cluster bis eine bestimmte Clusteranzahl erreicht ist. Beispiele für hierarchische Verfahren sind BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [Zhang et al. 1996], CURE [Guha et al. 1998].
- *Partitionierende Verfahren*: Mit partitionierenden Clustern wird der Merkmalsraum in eine vorgegebene Anzahl von Bereichen unterteilt. Die initiale Zuordnung wird durch den iterativen Ansatz schrittweise optimiert. Beispiele für Partitionierendes Clustering sind k-means [Wang & Hamilton 2003], PAM (Partitioning Around Medoids) [Kaufman, & Rousseeuw 1990], CLARA (Clustering for Large Applications) [Kaufman, & Rousseeuw 1990] und CLARANS (A Clustering Algorithm based on Randomized Search) [Ng & Han 1994].
- *Dichte-basierte Verfahren*: Mit dichte-basierten Verfahren werden Cluster als Gebiete mit dicht beieinander liegenden Objekten bezeichnet. Diese Gebiete werden von Gebieten mit weniger dicht liegenden Objekten getrennt. Beispiele für dichte-basierten Verfahren sind DBSCAN (Density Based Spatial Clustering of Applications with Noise) [Ester et al. 1996] und DENCLUE (DENsitybased CLUstEring) [Hinneburg & Keim 1998].

- *Weitere Verfahren*: Grid-basierte Verfahren quantisieren den Merkmalsraum in eine endliche Anzahl von Zellen und führen Operationen auf diesen Zellen aus. Die Zellen mit mehr als einem Punkt werden als dicht bezeichnet und die aufeinander folgenden Zellen werden als ein Cluster verbunden. Die Verfahren besitzen eine hohe Verarbeitungsgeschwindigkeit, da nicht die Anzahl der Objekte, sondern die Anzahl der Zellen eine Rolle spielt. Beispiele für Grid-basierte Verfahren sind STING (STatistical INformation Grid) [Wang et al. 1997] und CLIQUE (CLustering In QUEst) [Agrawal et al. 2005]. Neuronale Netze werden speziell in Form von Self-Organizing Feature Maps (SOMs) zum Clustering eingesetzt. SOM stellen keine eigenen Clusterverfahren dar, sondern sind Abbildungen vorhandener Verfahren auf einen diskreten 2D oder 3D Raum. Da SOM in der vorliegenden Arbeit verwendet wird, wird die Literatur zu SOM in einem separatem Kapitel vorgestellt.

Die Arbeit von [Ester et al. 1996] vergleicht die Effektivität und die Effizienz des partitionierenden Verfahrens CLARANS mit dem dichte-basierten Verfahren DBSCAN. CLARANS ist die erste Cluster-Methode für Spatial Data Mining, während das DBSCAN-Verfahren sich als die erste dichte-basierte Methode für Spatial Data Mining aufweist. Die Festlegung der Cluster verfolgt das Ziel, dass die Punktdichte innerhalb eines Clusters höher als außerhalb des Clusters ist. Ein anderes dichte-basiertes Cluster-Verfahren DBRS (Density-Based clustering with Random Sampling) wird in der Arbeit von [Wang & Hamilton 2003] vorgestellt.

Die CLARANS- sowie DBSCAN-Verfahren basieren auf einer diskreten raumbezogenen Struktur. Die Arbeit von [Sander et al. 1998] erforscht ein Cluster-Verfahren für willkürliche Formen von erweiterten raumbezogenen Daten (Linien und Fläche). Das Verfahren nennt sich GDBSCAN (Generalized Density Based Spatial Clustering of Applications with Noise). Es wird aus dem DBSCAN-Verfahren in zwei wichtigen Wegen generalisiert. Erstens, der Begriff Nachbarschaft kann weiter auf symmetrische und reflexive Objekte (z.B Polygon) verwendet werden. Zweitens, die nicht raumbezogenen Attribute können als ein Objekt in der Nachbarschaft verwendet werden. Mit beiden Wegen kann das generalisierte GDBSCAN-Verfahren sowohl punktförmige Objekte als auch erweiterte raumbezogene Objekte analysieren. In der Arbeit wird die Anwendung des Verfahrens noch auf 2D Punkte (Astronomie), 3D Punkte (Biologie), 5D Punkte (Erdwissenschaft) und 2D Polygone (Geographie) diskutiert.

Die räumliche Beziehung der Objekte wird in einem Graph-basierten Clustering von [Malerba et al. 2005] berücksichtigt. Die Cluster vom vorgestellten Verfahren CORSO (Clustering of Related Structured Objects) werden dort konstruiert, wo die strukturierten räumlichen Objekte nach ihrer räumlichen Beziehung und ihrer strukturellen Ähnlichkeit angeordnet werden. CORSO verwendet die multirelationale Methode für die Modellierung der Homogenität der Daten-Relation und minimiert mittels Nachbarschafts-Graphen die relationale Beschränkung an der Kante.

Ein Beispiel für Hierarchisches Clustering kann in [Anders 2003] gefunden werden. Das Verfahren wird als hierarchisches parameterfreies Graph-Clustering (HPGCL) bezeichnet. Die Graphenhierarchie der Nachbarschaft wird für die Definition der Nachbarschaft gebildet und die Ähnlichkeitskriterien der Cluster werden parameterfrei definiert.

3.2.2 Klassifizierung

Bei der Klassifizierung werden Objekte anhand der Attribute zu einer Klasse aus einer vorgegebenen Menge von Klassen zugeordnet.

Die Arbeit von [Li & Claramunt 2006] bringt ein Verfahren der Erweiterung der Anwendung des Entscheidungsbaums (Decision Tree) auf georeferenzierte Daten. Der Entscheidungsbaum wird in zahlreichen Bereichen angewendet und ist ein effizienter Algorithmus für die Klassifikation von großen Datenbanken. Die Konstruktion des Entscheidungsbaums geht von der Wahl eines Attributs für den Wurzelknoten aus und eine Verzweigung wird für jeden Attributwert mit neuem Knoten hinzugefügt. Die Testdaten werden nach Attributwerten abgefragt. Wenn alle Daten im neuen Knoten das gleiche Blatt (Klasse) haben, wird der Prozess beendet. Falls nicht, wird ein neues Attribut gewählt und der Vorgang wird wie beschrieben fortgesetzt. In der Arbeit wird der Einfluss der räumlichen Dimension beim Entscheidungsbaum berücksichtigt. Der Koeffizient der räumlichen Vielfalt wird in einem Entscheidungsbaum integriert. Die landwirtschaftlichen Daten Chinas werden mittels des Verfahrens untersucht. Das Verfahren wird für die Klassifikation großer georeferenzierter Datenbanken als effizient bewertet.

In der Arbeit von [Devadas et al. 2012] wird das objektbasierte SVM-Verfahren mit einem herkömmlichen pixelbasierten Klassifikationsverfahren Maximum-Likelihood-Classification (MLC) verglichen. Die Idee der Support Vector Machines ist die Unterteilung der Objekte durch eine Hyperebene in zwei Klassen. Eine Menge der linear trennbaren Trainingsobjekte wird ausgewählt, um eine Hyperebene zu wählen, wobei ein möglichst breiter Rand um die Klassen herum erhalten werden soll. Bei der MLC handelt es sich um eine Methode, die auf Wahrscheinlichkeitsdichtefunktionen basiert. Die Zuordnung eines Pixels zu einer Referenzklasse basiert auf der Basis des Mittelwertvektors und der Varianz-Kovarianzmatrix. Die Linien mit gleicher Zuordnungswahrscheinlichkeit werden in verschiedenen Merkmalsräumen ermittelt und die Zuordnung eines Pixels erfolgt nach dem Prinzip der größtmöglichen Wahrscheinlichkeit. In der Arbeit werden die Klassifizierungsverfahren in der Überwachung der Landnutzung angewendet. Die Untersuchungsfläche wird jeweils in zwei Jahreszeiten, und zwar Sommer 2010 und Winter 2011 als Trainingsdaten eingeteilt. Das Ergebnis des SVM (Support Vector Machine) wird mit dem des MLC verglichen: Die Klassifikation mit dem SVM-Verfahren enthält eine Genauigkeit von 95% und die mit dem herkömmlichen MLC-Verfahren enthält eine Genauigkeit von 89%.

Die Klassifizierung in Spatial Data Mining wird in der Arbeit von [Heinzle & Sester 2004] eingesetzt. Um die implizite Information aus räumlichen Datensätzen abzuleiten, wird eine räumliche Suchmaschine realisiert. Eine automatische Erläuterung der Datensätze wird mithilfe von wichtigen Metadata-Tags beschrieben. Metadata sind strukturierte Daten und ermöglichen dem Nutzer die Selektion und die Bewertung der Daten. Dies ist jedoch nur für Menschen interpretierbar. Um dem Computer die Lernfähigkeit der räumlichen Ausdrücke beizubringen, werden die Sprache- und Semantik-Translation durchgeführt. Die impliziten Informationen können auf zwei Wegen extrahiert werden. Zum einen werden Assoziierungsregeln definiert und auf die Daten eingesetzt. Zum anderen findet der Computer selbst die Regeln bei der Untersuchung der Daten. Beide Wege sind bekannt als Klassifizierungsmethoden des Data Mining.

3.3 Künstliche Neuronale Netze

Für die Klassifizierung der Kartenobjekte werden die künstlichen neuronalen Netze eingesetzt. Künstliche neuronale Netze sind sehr leistungsfähige Berechnungsmodelle. Sie werden in den letzten Jahren weitgehend für die Klassifizierungsaufgaben, nicht-lineares Mapping und Mustererkennung angewendet. Aufgrund ihrer Lernfähigkeit sind neuronale Netze in ihrem Gebrauch effizient. Im Folgenden wird die Literatur, in der neuronale Netze angewendet werden, in den signifikanten Forschungsgebieten vorgestellt.

[Schleinkofer 2007] verwendet das virtuelle neuronale Netz, um die Bauteile von Bauobjekten zu erkennen. Die neuronalen Netze erlernen durch Beispiele die implizit formulierten Regeln und wenden diese auf neue Objekte an. Somit werden die Bauteile klassifiziert. Räumliche Ausdehnung, Körperoberfläche, die Verschneidung der Objekte sowie die Objektkoordinaten werden als Eingangsparameter ausgewählt. Aus den Eingangsparametern werden vier Eingabeneuronen der neuronalen Netze festgelegt:

- x / y: Verhältnis der größeren zur kleineren horizontalen Ausdehnung.
- x / z: Verhältnis der größeren horizontalen zur vertikalen Ausdehnung.
- „liegt innerhalb von“: Anzahl an Objekten, innerhalb deren der aktuelle Körper liegt.
- Z-Abstand: Abstand der minimalen z-Werte des aktuellen Objektes und zu den umschließenden Objekten (sofern vorhanden transformiert, sonst zu 0 gesetzt). Der z-Abstand wird aus den Objektschwerpunkten sowie den -ausdehnungen in z-Richtung berechnet.

Die Bauteile bestehen aus sieben Klassen: Wand, Tür, Fenster, Decke, Deckendurchbruch, Stütze und Träger. Die sieben Klassen sind die Ausgabenneuronen in den neuronalen Netzen. Zwischen den Eingabeneuronen und Ausgabenneuronen wird eine Schicht mit sieben verdeckten Neuronen angeordnet (Abbildung 3-4).

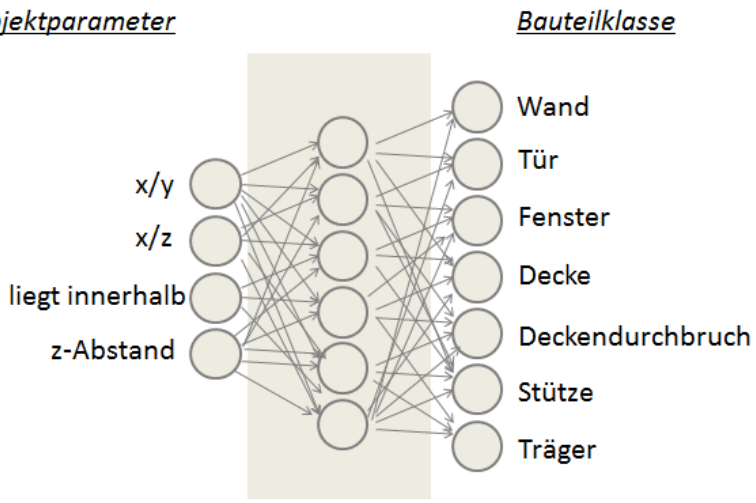


Abbildung 3-4: zweischichtiges Netz (nach [Schleinkofer 2007])

[Liu et al. 2005] leiten aus der Analyse der neuronalen Netze mit ART-MMAP, die eine Erweiterung von ARTMAP (vorhersagende Adaptive Resonanztheorie) aufweist, eine räumliche und zeitliche Data-Mining-Methode ab. Die Methode dient zur Simulation und zur Prognose der städtischen Ausdehnung. Die städtischen Eigenschaften am Beispiel von Verkehrswesen, Landnutzung und Topographie werden als Eingangsdaten der Neuronalen Netze verwendet. Die trainierten Netze werden dann auf die Untersuchungsregionen angewandt.

Die neuronalen Netze finden häufig Eingang in die Satellitenbilderkennung. Aus den Satellitenbildern wird die thematische Karte durch Klassifizierung abgeleitet. In der Arbeit von [Carvajal et al. 2006] wird zum Beispiel eine Methode zur Erkennung der Klasse Gewächshäuser basierend auf künstlichen neuronalen Netzen untersucht. Die radiometrischen und Wavelet-textuellen Eigenschaften werden dem Trainingsmodell zur Klassifizierung gelehrt.

Eine häufige Anwendung finden die neuronalen Netze in der medizinischen Bilderkennung z.B. in der Arbeit von [Karkanis et al. 2000]. Der Einsatz Neuronaler Netze im Bereich medizinischer Bildverarbeitung wird auch in den Arbeiten von [Coppini et al. 1995], [Hanka et al. 1996], [Ifeachor & Rosen 1994], [Innocent et al. 1997], [Karkanis et al. 1999], [Phee et al. 1998], [Veropoulos et al. 1998], [Zhu & Yan 1997] untersucht.

SOM

Da die SOM in der vorliegenden Arbeit für die Interpretation verwendet wird, wird im Folgenden auf deren Referenzen näher eingegangen.

In [Jardin & Séverin 2011] findet die SOM Anwendung in der Finanzwelt, indem sie den Prognosezeitraum eines finanziellen Zusammenbruchsmodells erweitert. Die Ergebnisse mittels SOM sind im

Vergleich zu den herkömmlichen Verfahren zu solchen Aufgaben wie z.B. Diskriminanzanalyse, logistische Regression oder Ereigniszeitanalyse stabiler. Ein anderes Beispiel für die Anwendung der SOM im Bereich der Finanzwelt ist in [Eklund 2002] zu finden.

[Brocki 2007] entwickelt einen Ansatz mit der SOM, um ein Problem des Handlungsreisenden (TSP) zu lösen. Rundreiseprobleme bestehen in der Wahl der Reihenfolge der zu besuchenden Orte, so dass die gesamte Reisestrecke des Reisenden nach der Rückkehr zum Ausgangsort am kürzesten ist. Das Problem des Handlungsreisenden wird auch von [Garcia & Moreno 2005] mittels einer verbesserten SOM analysiert.

Die SOM wird in [Frey 2012] auf die kontinuierliche Überwachung von Industrieprozessen eingesetzt, um frühzeitig potentielle Fehler und Betriebsstörungen zu erkennen. [Whigham 2005] untersucht die Anwendung der SOM in der Modellierung des ökologischen Systems. Auf die Modellierung von Geschäftsprozessmodellen wird die SOM in [Pütz & Sinz 2010] eingesetzt. In [Lampinen & Kostiainen 2000] wird die SOM in der Datenanalyse und Visualisierung der mehrdimensionalen Daten eingesetzt. Ferner findet die SOM in [Lichodzijewski et al. 2002] ihre Anwendung in einem Host-basierten Angriffserkennungssystem, um Netzwerke zu überwachen und die Sicherheit zu erhöhen. [Nuernberger & Detyniecki 2006] präsentieren ein auf SOM basiertes Verfahren für die Organisation und Klassifizierung von Email.

SOM in GIS

Es existieren auch zahlreiche Anwendungen der SOM mit geografischen Daten. Beispielsweise wird in [Sester 2007] ein auf SOM basiertes Programm TYPIFY vorgestellt. Das TYPIFY wird für die Generalisierung von Gebäudegrundrissen für kleinere Maßstäbe entwickelt. Die Anzahl der Gebäude wird reduziert, indem die Gebäude nach der originalen räumlichen Verteilung erneut angeordnet werden. Außerdem werden die kleinen Gebäude durch Quadrate ersetzt, während größere Gebäude die ursprüngliche Gestalt behalten. Die erneute räumliche Verteilung der Quadrate wird mittels SOM realisiert.

Die Arbeit von [Fincke et al. 2008] untersucht die Möglichkeit, die herkömmlich aus nicht räumlichen Daten abgeleitete SOM in GIS zu importieren, so dass die Operationen für die räumlichen Analysen auch für die ursprünglich nicht räumlichen Daten verwendet werden können. In [Spielman & Thill 2008] wird die SOM eingesetzt, um die geodemographische Klassifizierung zu erweitern. In [Bação et al. 2005] werden neben der Standard-SOM drei für die georeferenzierten Daten geeignete SOM-Varianten vorgestellt: Hierarchische SOMs, Geo-enforced SOM und Geo-SOM.

3.4 Webcrawler

Die Suche nach den Vektorkarten in der vorliegenden Arbeit basiert auf Webcrawler, welcher im Bereich der Suchmaschinen bereits seit längerem erfolgreich eingesetzt wird. Nachfolgend wird einige Literatur über Webcrawler vorgestellt.

Webcrawler findet eine häufige Anwendung in der Suche nach Bildern im Web. Ein s.g. Bild-Webcrawler kann nach [Khurana & Kumar 2012a] in zwei Typen aufgeteilt werden:

- *Inhaltsbasierter Bild-Webcrawler:* Die Bildeigenschaften wie z.B. Farbe und Helligkeit werden bei der Suche nach Bildern analysiert.
- *Stichwortbasierter Bild-Webcrawler:* Die Textbeschreibung der Bilder wird bei der Suche nach Bildern berücksichtigt.

Der stichwortbasierte Bild-Webcrawler wird bereits häufig erforscht. In [Rowe 2002] wird ein Tool, ein textbasierter Webcrawler, beschrieben. [Khurana & Kumar 2012a] stellen fest, dass ihr textbasierter Bild-Webcrawler mit hoher Genauigkeit und Geschwindigkeit funktionieren kann. Im Vergleich zu stichwortbasierter Bildsuche wird bei einer inhaltbasierten Bildsuche mehr Zeit benötigt, um die Bildverarbeitung durchzuführen. [Ren et al. 2002] legen den Schwerpunkt für ein Bildsuchsystem auf den räumlichen Kontext. Die Anzahl der Objekte, die Fläche und die räumliche Beziehung werden dabei berücksichtigt. In [Jain et al. 2013] wird basierend auf der Kombination der Farbe, Struktur und Form der Bilder eine effiziente Bildsuche entwickelt.

Wenn ein Webcrawler die Webseiten nach spezifischen Themen durchsucht, dann wird er als fokussierter Webcrawler bezeichnet. Die Suche nach Webseiten vorgegebener Themen ist kein einfacher Job und wird in zahlreicher Literatur untersucht. Beispielsweise entwickeln [Liu et al. 2008] ein Framework auf der Basis von MEMMs (Maximum Entropy Markov Models). Der fokussierte Webcrawler extrahiert verschiedene Features aus den Webseiten wie beispielsweise den Ankertext und Stichwörter in URL, um die nützlichen Kontexte repräsentieren zu können. In [Li et al. 2009] wird eine Methode zu zwei Problemen des fokussierten Webcrawlers diskutiert. Bei einem geht es um die Definition des spezifischen Themas. Beim anderen handelt es sich um die effiziente Sortierung der heruntergeladenen Links.

In [Khurana & Kumar 2012b] wird der Webcrawler verbessert, um das (nahe) Duplikat der Webseiten zu vermeiden. [Narayan Das & Kumar 2012] präsentieren das Verfahren HEET (Hidden Web Query Technique) für die Modellierung und Abfrage des versteckten Webs. Eine andere Verbesserung des Webcrawler, die Reduzierung des Internet-Traffics, wird in [Mishra et al. 2010] beschrieben. Webcrawler kennt die Aktualisierung der Webseiten, so dass weniger Abfragen für die aktualisierten Webseiten notwendig sind. Dafür soll die Liste der aktualisierten URLs in einer auf HTML basierten

Datei gespeichert und dem Root-Verzeichnis von Websites platziert werden. Somit muss der Webcrawler nicht in der ganzen Website suchen.

4 Aufbereitung der Karten

Um Karten hinsichtlich ihrer Semantik zu interpretieren, ist es notwendig, diese für die Untersuchung auszuwählen und aufzubereiten. In diesem Kapitel wird erläutert, welche Kartensätze wie bereitgestellt werden.

Die Karten in der Arbeit stammen aus der größten elektronischen Datenbank der Welt, dem World Wide Web. Mit der Entwicklung des Webs werden hier in den letzten Jahren vermehrt Geodaten dargeboten, die als Datenquelle der vorliegenden Arbeit dienen sollen. Die Karten sind in unterschiedlichen Formaten, Koordinatensystemen, Maßstäben und Datentypen vorhanden. Es gibt sowohl Karten, die eigens für das Internet angefertigt werden, als auch Karten, die in Formaten vorliegen, die nicht in den Standardbrowsern angesehen werden können. Die digitalen Karten werden in Raster- und Vektorkarten gegliedert. Im Folgenden wird zunächst diskutiert, ob Rasterkarten oder Vektorkarten als Datenquelle für diese Arbeit verwendet werden sollen.

4.1 Rasterkarten vs. Vektorkarten

Digitale Karten lassen sich in Rasterkarten und Vektorkarten differenzieren. Bei Rasterkarten werden die Karten als Bild abgespeichert. Sie sind mit einer eingescannten Papierkarte vergleichbar und bestehen aus einer rasterförmigen Anordnung von Pixeln, welchen ein Farb- oder Grauwert zugewiesen wird. Vektorkarten hingegen sind digitale Karten, deren Bild über mathematische Funktionen erzeugt wird und die aus Vektoren aus einer Datenbank aufgebaut sind.

4.1.1 Rasterkarten

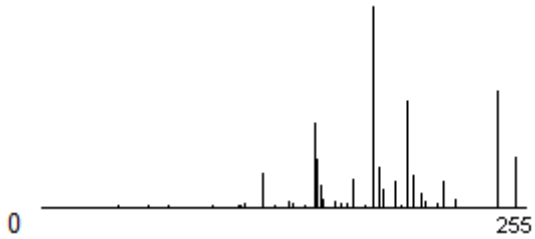
Rasterkarten sind im Grund genommen Bilder und werden in Bildformaten der Endungen .img, .png, usw. gespeichert. Um die Rasterkarten im Internet zu finden, kann die Bildersuchfunktion von Suchmaschinen wie Google verwendet werden. Die Abbildung 4-1 zeigt einen Teil der erhaltenen Suchergebnisse der Google-Bildersuche nach dem Stichwort „map“.

Abbildung 4-1: Teilergebnisse der Bildersuche nach „map“

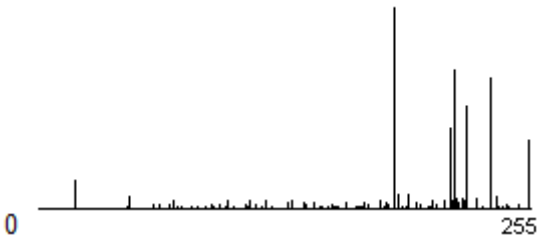
Die Ergebnisse bestehen zum großen Teil aus Rasterkarten. Es sind jedoch auch Nicht-Rasterkarten dabei. In den 136 Beispielergebnissen sind 39 Nicht-Rasterkarten enthalten. Um die Rasterkarten von anderen Bildern zu trennen, soll die Interpretation der Rasterkarten automatisch realisiert werden. Die Trennung von Rasterkarten und Fotos kann auf zwei verschiedene Weisen erreicht werden. Zum einen über die Prüfung der Homogenität, zum anderen über das Histogramm.

In den meisten Karten existieren größere homogene Gebiete, in denen identische Farbtöne oder Grauwerte vorliegen. Ein Foto dagegen hat durch seine Eigenschaften der Bildaufnahme wie Belichtung, Kontrast und Dynamik nur sehr kleine homogene Gebiete.

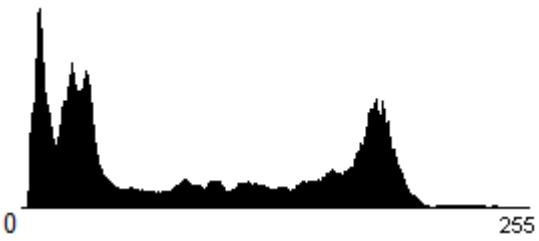
Auch ein Histogramm, das die Häufigkeitsverteilung der vorhandenen Tonwerte eines Bildes anzeigt, kann Rasterkarten und Fotos unterscheiden helfen. Aufgrund der Eigenschaften der Bildaufnahme erhalten Fotos deutlich mehr Tonwerte als Rasterkarten. Die Abbildung 4-2 stellt Beispiele des Histogramms für Rasterkarten sowie Fotos gegenüber. Ein Histogramm wird in einem zweidimensionalen Koordinatensystem dargestellt: Die horizontale Achse fängt links bei schwarz mit dem Tonwert null an und endet rechts bei weiß mit Tonwert 255. Die Höhe der Kurve entspricht der Pixelmenge der jeweiligen Tonwerte. Dabei ist die Verteilung der Pixel interessant, die absolute Zahl spielt keine Rolle.



(1)



(2)



(3)

(4)

Abbildung 4-2: Beispielhistogramme für Rasterkarten (1), (2) sowie Fotos (3), (4)

An den gezeigten Beispielen ist zu erkennen, dass in Fotos alle Tonwerte vorhanden sind, während Rasterkarten nur eine begrenze Anzahl von Tonwerten enthalten. Die Rasterkarten lassen sich anhand der Homogenität und anhand des Histogramms von Fotos differenzieren. Eine Unterscheidung von Bildern wie die in der Abbildung 4-3 dargestellten Diagramme ist jedoch nicht effizient, da solche Grafiken sowohl im Hinblick auf Homogenität als auch auf Histogramme sehr ähnliche Eigenschaften wie Rasterkarten haben.

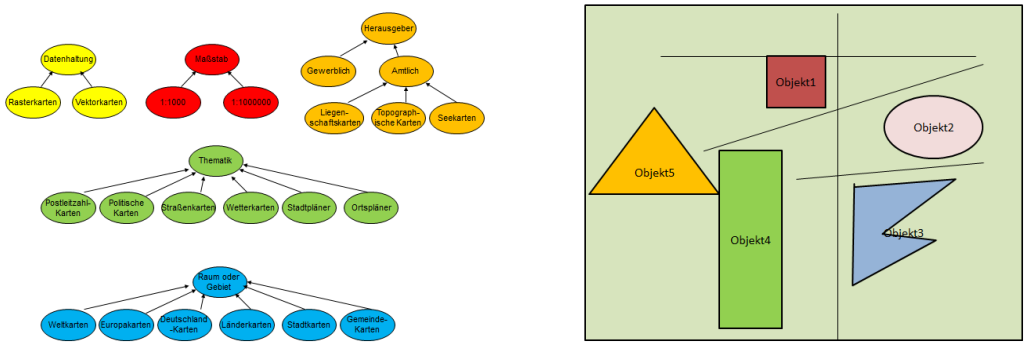


Abbildung 4-3: Beispieldiagramme

4.1.2 Vektorkarten

Im Gegensatz zu Erkennung der Rasterkarten, ist die Erkennung einer Datei als Vektorkarte einfacher realisierbar. Dem liegt zugrunde, dass eine Vektorkarte mit einem speziellen Format gespeichert wird. Um Vektorkarten im Internet aufzufinden, muss lediglich das entsprechende Format gesucht werden. In dem nachfolgenden Unterkapitel soll näher beschrieben werden, wie Vektorkarten aus dem Internet aufgegriffen werden.

In der vorliegenden Arbeit soll nach Vektorkarten des Formats ESRI-Shapefile gesucht werden. Dies bietet sich an, da das Shapefile-Format ein weit verbreitetes Format ist. Dank der niedrigen Komplexität und der hohen Datenqualität hat sich das Shapefile mittlerweile zu einer Art Standarddatei im GIS-Umfeld entwickelt. Infolgedessen ist eine fast unbegrenzte Menge an Shapefiles im Internet zugänglich. Das Problem dabei ist jedoch, die irrelevanten Daten herauszufiltern und die Shapefiles aus den enorm vielen Informationen zu identifizieren.

Mittlerweile existieren viele Internetsuchmaschinen, mit denen Objekte durch Suchbegriffe gesucht werden können. Die Suche nach den Shapefiles durch einen Suchbegriff beispielsweise „Shapefile“ ist jedoch ineffizient. Der Begriff „Shapefile“ wird zwar wörtlich von der Suchmaschine verstanden, jedoch kann die Suchmaschine nicht erkennen, dass in diesem Fall nach Shapefiles gesucht werden soll. Die Suchmaschine Google gibt auf eine einfache Suche nach dem Begriff „Shapefile“ sehr schnell zahlreiche Treffer an, von welchen aber nur eine kleine Anzahl gewünschte Resultate sind. Abbildung

4-4 illustriert, dass viele Webseiten mit der Erläuterung des Begriffs „Shapefile“ als Resultat ausgegeben werden. Das Herausfiltern der gewünschten Shapefiles aus der großen Anzahl der Resultate ist sehr aufwendig.

Abbildung 4-4: Teilergebnisse der Suche nach „Shapefile“

Um genauere Suchergebnisse zu erreichen, können viele Suchmaschinen auf bestimmte Suchkriterien eingeschränkt werden. Google bietet z.B. die Möglichkeit, nach bestimmten Dateitypen wie z.B. pdf, ppt, xls usw. zu suchen. Eine auf den Shapefile beschränkte Suche existiert jedoch noch nicht. Aus diesem Grund wird in dieser Arbeit eine Suchmaschine entwickelt, um Shapefiles im Internet suchen zu können.

Für eine wirkungsvolle Suche der Shapefiles ist es notwendig, sich mit der Struktur des Formats auseinanderzusetzen. Ein Shapefile ist kein einzelner Dateityp, es kann aus Dateien mit den Endungen .shp, .dbf und .shx, häufig auch noch aus den optionalen Dateien wie z.B. Dateien mit den Endungen atx, .sbx, .prj und .cpg etc. bestehen. Ein Shapefile wird im Normalfall komprimiert, so dass die verschiedenen Dateien einer Vektorkarte in einem komprimierten Ordner zusammengefasst werden können. Zudem ist ein Shapefile oftmals riesig. Durch das Komprimieren wird weniger Speicherplatz benötigt und die Daten können schneller übertragen werden.

Im folgenden Unterkapitel wird erläutert, wie eine webcrawler-basierte Suchmaschine entwickelt wird, um Shapefiles im Internet ausfindig zu machen.

4.2 Webcrawler

Für die Suche der Shapefiles im Internet wird der Webcrawler eingesetzt. Shapefiles liegen in komprimierter Form vor. Aus diesem Grund wird nach komprimierten Dateien mit .zip-Endung gesucht. Die gefundenen Dateien werden lokal dekomprimiert und nach dem Dateityp verifiziert. Falls es sich um ein Shapefile handelt, wird die jeweilige Datei als Kandidat zur Untersuchung der Karteninterpretation genutzt.

Der Webcrawler in der vorliegenden Arbeit wird mit der Java-Programmiersprache implementiert. Die Implementierung basiert auf einem Open Source Webcrawler Framework *webmagic*. Webmagic ist ein Webcrawler Framework für eine benutzerdefinierte Entwicklung. Webmagic folgt dem Robots-Exclusion-Standard-Protokoll, d.h. webmagic sucht nur die Webseiten, die frei zur Verfügung gestellt werden (vgl. 2.1.2). Mit webmagic wird eine einfache flexible API angeboten und es ist hierbei keine Konfiguration erforderlich [Huang 2013]. Die Funktionalitäten des webmagics beinhalten den gesamte Lifecycle eines Webcrawlers: *Downloader*, *PageProcessor*, *Scheduler* und *Pipeline* (Abbildung 4-5):

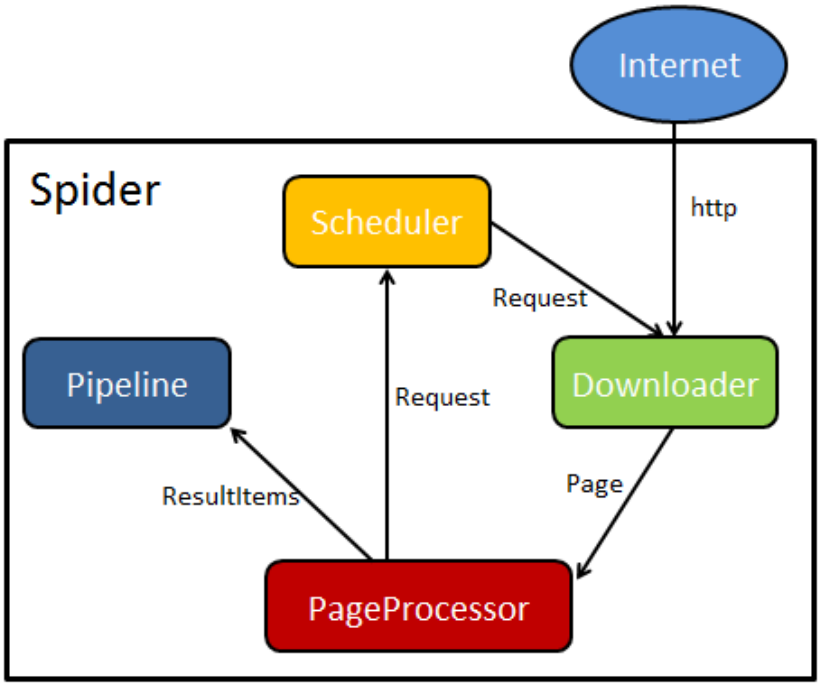


Abbildung 4-5: Lifecycle des webmagics (nach [Huang 2013])

- *Downloader.* Mit dem *Downloader* werden die Webseiten heruntergeladen. Das Herunterladen ist der Anfang der Arbeit eines Webcrawlers. Dies wird mit einem Http-Request umgesetzt.
- *PageProcessor.* Nach dem Herunterladen werden die Webseiten mit dem *PageProcessor* nach URLs geparst. Webmagic bietet eine eigene Komponente - den *Selector* - an, mit dem das Parsen einfach umgesetzt werden kann.
- *Scheduler.* Mit dem *Scheduler* werden die URLs verwaltet. Die nicht indexierten URLs und die indexierten URLs werden getrennt gespeichert. Dabei wird eine Wiederholung der URLs vermieden.
- *Pipeline.* Mit *Pipeline* werden die Ergebnisse exportiert und gespeichert.

Auf webmagic aufbauend wird ein Webcrawler in der Arbeit entwickelt. Webmagic unterstützt die Suche mit der Nebenläufigkeit, d.h. mehrere Threads können als einzelne, unabhängige, für sich laufende Webcrawler arbeiten. Die Suche für den Webcrawler wird mit drei Threads durchgeführt, um die Leistung zu erhöhen. Die Ergebnisse werden nicht in der Komponente Pipeline von webmagic, sondern in einer MySQL-Datenbank gespeichert. So können die Ergebnisse besser verwaltet und abgefragt werden.

Der Webcrawler wird in zwei Ansätzen untersucht. Beim Ersten handelt es sich um eine Suche ohne jegliche Einschränkungen. Beim Zweiten findet die Suche lediglich in einem Server statt. Die zu durchsuchenden Server werden vorher durch Google-Suche festgelegt. Nachfolgend werden beide Ansätze vorgestellt.

4.2.1 Vollautomatisch

Bei der ersten Untersuchung wird der Link „http://www.ifp.uni-stuttgart.de“ als Start-URL definiert. Der Webcrawler startet mit diesem Link und durchsucht URLs im Internet. Die indexierten URLs werden in eine *url*-Tabelle in einer MySQL-Datenbank gespeichert. Die *url*-Tabelle verfügt über drei Spalten: ID, URL und die Empfangszeit (Abbildung 4-6). Die URLs werden auf die Endung .zip geprüft. Liegt ein URL mit der Endung .zip vor, so wird der URL in die *zip*-Tabelle in der Datenbank gespeichert.

	id	url	get_time
	1	http://www.adobe.com/products/acrobat/readstep.htm...	2011-09-19 03:21:58
	2	http://www.oregon.gov/website_feedback.shtml	2011-09-19 03:21:58
	3	http://www.oregon.gov/sitemap.shtml	2011-09-19 03:21:58
	4	http://www.oregon.gov/DAS/EISPD/EGOV/termsconditio...	2011-09-19 03:21:58
	5	http://oregon.gov/DAS/fileformats.shtml	2011-09-19 03:21:58
	6	http://www.oregon.gov/index.shtml	2011-09-19 03:21:58
	7	http://www.oregon.gov/accessibility.shtml	2011-09-19 03:21:58
	8	http://transcoder.usablenet.com/tt/referrer	2011-09-19 03:21:58
	9	http://www.oregon.gov/ODVA/VETFORM.shtml	2011-09-19 03:21:58
	10	http://www.oregon.gov/welcome.shtml	2011-09-19 03:21:58
	11	http://www.leg.state.or.us/ors/	2011-09-19 03:21:58
	12	http://arcweb.sos.state.or.us/banners/rules.htm	2011-09-19 03:21:58
	13	http://www.oregon.gov/a_to_z_listing.shtml	2011-09-19 03:21:58
	14	http://www.oregon.gov/state_contact.shtml	2011-09-19 03:21:58

Abbildung 4-6: die url-Tabelle

Die Suchmaschine ist mit einem Intel(R) Core(TM)2 i7-3630QM-Computer (2,4GH_z 8GB RAM) sowie der Internetgeschwindigkeit 12.634 kbit/s ca. vier Tage gelaufen, dabei werden 3.000.012 Urls und 1,080 .zip-Dateien gefunden. Aus den .zip-Dateien können keine Shapefiles identifiziert werden. Die Leistung sowie die Effizienz der Suche nach Shapefiles sind sehr niedrig. Um die Leistung zu erhöhen, soll der Webcrawler denjenigen Server suchen, bei denen wahrscheinlich Shapefiles vorhanden sind. Im nächsten Abschnitt wird die Suche mit Einschränkungen betrachtet.

4.2.2 Mit Google-Unterstützung

Eine Liste von Servern, in welchen möglicherweise Shapefiles existieren, wird zunächst mittels Google gesucht. Dazu wird in Google-Suche „Shapefile download“ eingegeben. Es werden 635.000 Ergebnisse geliefert. Die ersten 80 Server werden einer Liste hinzugefügt und der Webcrawler wird so angepasst, dass er lediglich einen Server aus dieser Liste durchläuft. Er geht nicht über den Server hinaus und fragt andere Server im Internet nicht ab. Die indexierten URLs werden bei der Suche auf ihre Endung geprüft, die URLs mit der .zip-Endung werden in die Datenbank gespeichert.

Die .zip-Dateien werden durch das Abfragen der URLs aus der Datenbank aus dem Internet heruntergeladen und entpackt. Es wird erkannt, dass es sich bei jeder .zip-Datei um ein Shapefile oder mehrere Shapefiles handelt. Aus den 80 Servern werden insgesamt 10018 Shapefiles gefunden. In der folgenden Tabelle 4-1 wird ein Teil der von Google gefundenen Server sowie die Anzahl der gefundenen Shapefiles aufgelistet.

Server	Anzahl der Shapefile
downloads.cloudmade.com	4629
mapcruzin.com	564
vdstech.com	3
esri.de	0
mygeo.info	0
washington.edu	117
census.gov	1730
diva-gis.org	21
…	…

Tabelle 4-1: Beispielserver mit Shapefiles

4.2.3 Vergleich der Verfahren

Nun sollen die Ergebnisse der zwei vorgestellten Strategien verglichen werden. Die Suche des Webcrawlers ohne jegliche Einschränkungen startet mit einem Start-URL und findet im ganzen Internet statt. Die Suche mit Einschränkungen sucht nach URLs in bestimmten Servern, die durch Google mit dem Suchbegriff „shapefile download“ gefunden werden. Um beide Strategien zu vergleichen, wird die Anzahl der Shapefiles durch das Auffinden von jeweils ca. 200.000 URLs in der Tabelle 4-2 verglichen.

Strategie	Shapefiles	Trefferquote
Ohne Google-Suche	0	0.00%
Mit Google-Suche	5 786	2.89%

Tabelle 4-2: Vergleich der Ergebnisse

Aus der Tabelle lässt sich ablesen, dass die Suche mit Google-Unterstützung deutlich wirkungsvoller ist, wobei die Komplexität der Implementierung sowie die Suchgeschwindigkeit beider Strategien auf einer ähnlichen Stufe sind.

4.3 Abspeichern der Vektorkarten

Die Shapefiles sollen in einem Dateisystem gespeichert werden. Wie bereits erwähnt, existieren zu einem Shapefile im Normalfall außer der .shp-Datei, Dateien mit gleichem Namen jedoch unterschiedlicher Endung wie beispielsweise .shp, .dbf, .shx etc.. Alle Dateien mit identischem Namen werden in einem Ordner gespeichert. Der Ordner wird nach dem gemeinsamen Namen benannt. Wenn eine entpackte .zip-Datei mehrere Shapefiles enthält, werden alle zu einem Shapefile gehörigen Dateien in einem eigenen Ordner zusammengefügt. Insgesamt werden 10018 Ordner erzeugt.

4.4 Diskussion

Eine allgemeine Trennung von Rasterkarten zu sonstigen Bildern war nicht erfolgreich, da eine generelle Beschreibung von Rasterkarten sehr schwer zu definieren und die Darstellung der verschiedenen Bilder äußerst heterogen ist. In speziellen Fällen ist es denkbar, bestimmte Rasterkarten erkennen zu können, da Rasterkarten in verschiedenen Typen vorliegen und spezifische Eigenschaften als Charakteristika definiert werden. So können beispielsweise zur Erkennung von politischen Karten bestimmte Eigenschaften untersucht werden. Geographische Grenzen einer politischen Karte können durch das Verfahren zur Kantendetektion entdeckt werden. Die Grenzen besitzen meistens eine Netzform. Außerdem enthält jede politische Gliederung häufig Namen, die durch Texterkennungsverfahren erkannt werden können. Zudem können die Tonwerte der politischen Gliederungen analysiert werden. Sie werden oft mit unterschiedlichen Farben dargestellt.

Der Webcrawler in dieser Arbeit ist eine einfache Internet-Suchmaschine. Die Suchgeschwindigkeit des Webcrawlers ist niedrig. Für das Auffinden von 3.000.000 URLs werden ca. vier Tage benötigt. Die Geschwindigkeit hängt dabei von der Hardwareausstattung des Computers sowie der Bandbreite des Internets ab. Außerdem kann die Geschwindigkeit erhöht werden, indem die Anzahl der Threads erhöht wird oder einer der führenden Java-Frameworks für Internet-Suchmaschinen wie z.B. Nutch verwendet werden.

Die Google-Suche nach Servern, in denen Shapefiles wahrscheinlich zum Herunterladen zur Verfügung stehen, macht die Ermittlung der Shapefiles wesentlich effizienter. Diese vorauswählende Suche wird nicht in dem Webcrawler der vorliegenden Arbeit integriert, da Google viel schneller arbeiten kann. Die 635.000 Ergebnisse auf die Suchanfrage „Shapefile download“ werden in 0,24 Sekunden geliefert.

5 Interpretation des einzelnen Objekts

Nachdem die Beschaffung der Vektorkarten aus dem Internet erläutert wurde, soll nun auf die Karteninterpretation eingegangen werden. Es wird die Interpretation der einzelnen Objekte mit einem auf SOM basierenden Verfahren vorgestellt. Die SOM dient dazu, die menschliche Interpretation zu simulieren. Basis dabei sind digitale Vektorkarten. Sie enthalten geometrisch strukturierte Objekte, deren Eigenschaften die SOM nutzt.

Zunächst sollen die Datensätze aufbereitet werden, welche möglichst viele Differenzen zwischen den Objektklassen beinhalten. Werden die im Kapitel 4 beschriebenen Vektorkarten des Internets betrachtet, so wird festgestellt, dass ein Shapefile ausschließlich eine Art von Objekten enthält. Beispielsweise beinhaltet ein Shapefile jeweils lediglich Gebäudegrundrisse, Straßen, oder Flüsse etc.. Insofern sind Shapefiles für die Interpretation der Kartenobjekte nicht geeignet.

Damit möglichst viele verschiedene Objektklassen auf einer Datei existieren, werden die Testkarten auf Ausschnitte einer Rasterkarte digitalisiert und als Shapefile gespeichert. Die Rasterkarte ist ein Stuttgarter Stadtplan, der von der Landeshauptstadt Stuttgart unter http://www.stuttgart.de/stadtplan/html/de/1280x1024.html bereitgestellt wird. Die vektorisierte Karte (Abbildung 5-1) ist mit einem großen Erfassungsmaßstab von 1:10.000 abgebildet. Die Objekte auf der Karte werden durch Polygone repräsentiert. Der gewählte Ausschnitt liegt in einem ländlichen Gebiet, so dass die Objekte in die Klassen Gebäudegrundriss, Ackerfläche, Wald und Straßennetz eingruppiert werden.

Abbildung 5-1: Testkarte (Ausschnitt aus Oberaichen und Musberg)

Die Kategorisierung der einzelnen Objekte in Objektklassen ist Voraussetzung des Interpretationsprozesses dieser Arbeit. Ziel der Interpretation ist, die einzelnen raumbezogenen Objekte ihren entsprechenden Klassen automatisch zuzuordnen. Dafür ist es notwendig, die Objekte mit Attributen zu versehen. Für jede raumbezogene Objektklasse wird anhand ihrer geometrischen Merkmale ein standardisierter Parametervektor erstellt, der als Muster in der Trainingsphase gelernt wird, so dass die Objekte nach der Trainingsphase in SOM entsprechend ihrer Klassen erkannt werden können. Dazu ist die Berechnung der Eingabeparameter jedes Objektes nach dem Lernen notwendig. Die Eingabeparameter des Objektes werden in die SOM eingegeben und der entsprechenden Objektklasse zugeordnet. Im Folgenden werden einzelne Definitionen der Merkmale vorgestellt.

5.1 Merkmalsdefinition

Um die Unterschiede zwischen den Objektklassen zu verdeutlichen, werden sie durch verschiedene Merkmale beschrieben. Eine genaue Definition der Eingangswerte und eine geeignete Wahl der Parameter spielt eine wichtige Rolle in der Trainingsphase in SOM. Im Hinblick auf die gewählte Fläche werden die Parameter durch bestimmte geometrischen Eigenschaften der Objekte definiert. Die Nachbarschaftsbeziehungen zwischen den einzelnen Objekten werden hier nicht berücksichtigt, da sie in diesem Fall nicht aussagekräftig sind. Die geometrischen Eigenschaften sind dem Kriterium: Menschliches Auge abgeleitet. Geometrische Eigenschaften resultieren also daraus, was das bloße menschliche Auge mindestens grob erfassen kann:

- Die Rechtwinkligkeit.
- Der Umfang.
- Die Fläche.
- Die Breite
- Die Innenfläche, welche von Straßennetz eingeschlossen wird.

5.1.1 Die Rechtwinkligkeit

Die Gebäudegrundrisse weisen die Eigenschaften Rechtwinkligkeit und Geradlinigkeit von Gebäudekanten auf. Selbst wenn die Gebäudegrundrisse vereinfacht auf einer Karte dargestellt werden, müssen diese typischen Objektformen erhalten bleiben [Sester 2000]. Dagegen besitzen Objekte wie Straßen oder Ackerflächen diese Eigenschaft nicht. Die Rechtwinkligkeit ist dadurch gekennzeichnet, dass der Winkel zwischen zwei Objektkanten 90° beträgt. Wobei die Bestimmung des rechten Winkels mit einem Toleranzbereich 5° ausgestattet werden soll, da die Karte durch die manuelle Digitalisierung nicht hundertprozentig präzise ausgefertigt werden kann.

5.1.2 Der Umfang

Der Umfang ist eine wichtige Eigenschaft eines Polygons. Die Differenzen der Umfänge der Objektklassen sind signifikant. Beispielsweise ist der Umfang eines Waldes anschaulich größer als der eines Hauses. Die Polygone auf der Karte haben meistens keine reguläre Form wie Kreis, Dreieck oder Rechteck, so dass hier der Umfang nicht durch Formeln berechnet werden kann. Vielmehr werden sie aus mehreren freien Linien zusammengebaut. Daraus resultierend wird der Umfang durch die Summe alle Kantenlängen definiert.

Um die Objektklassen durch Umfang voneinander zu unterscheiden, ist ein statistischer Schwellenwert des Umfangs notwendig. Zu diesem Zweck wird der durchschnittliche Umfang aller Objekte der Testkarten berechnet. Die Plausibilität des Durchschnitts hängt davon ab, ob die Häufigkeitsverteilung des Umfangs normal verteilt ist. Die Abbildung 5-2 zeigt, dass die Häufigkeitsverteilung des Umfangs des Gebäudegrundrisses prinzipiell der Normalverteilung entspricht. Dies gilt ebenso für die übrigen Klassen Ackerfläche, Wald und Straßennetz.

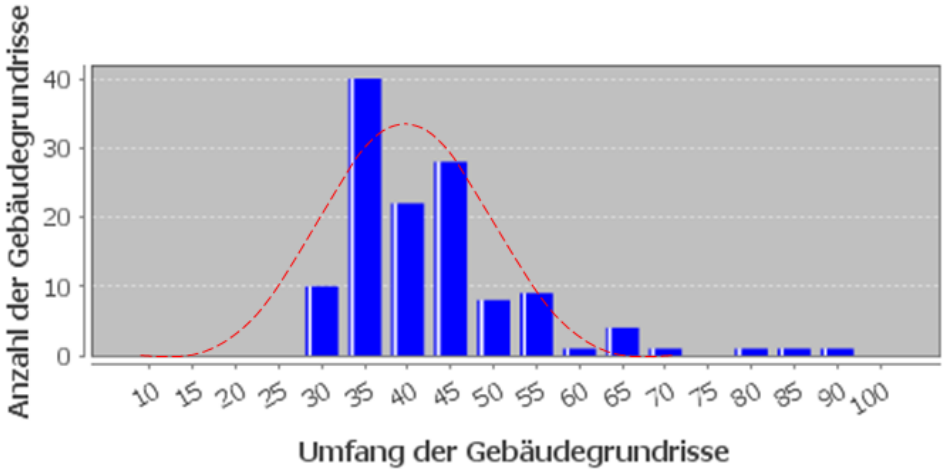


Abbildung 5-2: Die Häufigkeits- und Normalverteilung des Umfangs des Gebäudegrundrisses

Die folgende Tabelle 5-1 zeigt den durchschnittlichen Umfang jeder Objektklasse und den jeweils dadurch bestimmten Parameter. Da die Differenz der Durchschnitte deutlich ist, ist ein Schwellenwert praktisch zu determinieren. Der Schwellenwert 500 m trennt die Klassen in zwei Gruppen. Die eine Gruppe enthält die Klassen Gebäudegrundriss sowie Ackerflächen. Sie sind jeweils mit einem Umfang kleiner als 500 m definiert. Die andere Gruppe mit den Klassen Wald sowie Straßennetz ist jeweils mit einem Umfang größer als 500 m ausgestattet.

Klassen	Ø Umfang (m)	Parameter (m)
Gebäudegrundriss	39,95	Umfang < 500
Ackerfläche	207,17	Umfang < 500
Wald	1381,0	Umfang > 500
Straßennetz	6203,0	Umfang > 500

Tabelle 5-1: Der durchschnittliche Umfang und daraus bestimmter Parameter

5.1.3 Die Fläche

Die Fläche ist eine weitere wesentliche Eigenschaft der polygonförmigen Geodaten. Die Fläche eines Waldes ist beispielsweise deutlich größer als die eines Hauses. Die Fläche stellt geometrisch eine andere Perspektive als der Umfang dar. So hat eine Straße einen großen Umfang jedoch keine große Fläche. Um die Fläche der Polygone mit irregulären Formen zu berechnen, werden die Polygone in Teilpolygone zerlegt. Die Fläche entspricht demzufolge der Summe der Teilflächen.

Ähnlich des Vorgehens beim Umfang, wird auch bei der Fläche vorgegangen. Die Schwellenwerte basieren auf den in der Tabelle 5-2 veranschaulichten Durchschnittsflächen. Der Durchschnitt gilt als

plausibel, da die Häufigkeitsverteilung normal verteilt ist. Aus den Durchschnitten können drei Wertebereiche aus zwei Schwellenwerten abgeleitet werden. Die Fläche der Klasse Gebäudegrundriss ist kleiner als 1000 m², die Fläche der Klasse Ackerfläche liegt im Bereich von 1000 m² bis 10.000 m². Die Klassen Wald sowie Straßennetz besitzen jeweils Flächen größer als 10.000 m².

Klassen	Ø Fläche (m²)	Parameter (m²)
Gebäudegrundriss	128,218	Fläche < 1000
Ackerfläche	2283,29	1000 < Fläche < 10.000
Wald	99.044,30	Fläche > 10.000
Straßennetz	17.751.21	Fläche > 10.000

Tabelle 5-2: Die durchschnittliche Fläche und daraus bestimmter Parameter

5.1.4 Die Breite

Neben dem Umfang und der Fläche stellt die Breite eine weitere wesentliche Eigenschaft des Polygons dar. Das Straßennetz sowie der Gebäudegrundriss sind weniger breit, während die anderen Klassen über keine solche schmale Form verfügen. Die einfache Berechnung der Breite wird durch das Dividieren der Fläche durch den Umfang realisiert.

Wie bei Umfang und Fläche soll ein Schwellenwert für die Breite definiert werden. Der Durchschnitt (Tabelle 5-3) spielt bei der Ableitung des Schwellenwertes die bestimmende Rolle. Auch hier ist die Häufigkeitsverteilung normal verteilt. Zwei Wertebereiche können durch den Schwellenwert 5 m abgeleitet werden. Die Breite der Klassen Gebäudegrundriss und der Straßen im Straßennetz ist unter 5 m, die Breite der Klassen Ackerfläche und Wald liegt über 5 m.

Klassen	Ø Bereite (m)	Parameter (m)
Gebäudegrundriss	2,45	Breite < 5
Ackerfläche	11,02	Breite > 5
Wald	77,38	Breite > 5
Straßennetz	2,86	Breite < 5

Tabelle 5-3: Die durchschnittliche Breite und daraus bestimmter Parameter

5.1.5 Die Innenfläche

Durch die netzförmige Struktur von Straßen sind Innenflächen im Straßenpolygon vorhanden. Solche Flächen werden von Straßen vollständig eingeschlossen. In der Arbeit von [Anders 2007] werden sie als Straßenmaschen bezeichnet. In Siedlungsgebieten befinden sich mehr solcher Straßenmaschen als in anderen Gebieten. Die Innenfläche kann als ein weiterer Parameter definiert werden.

Die o.g. Parameter werden nun in einer Liste zusammengefügt. Wenn eine Klasse einen Parameter erfüllt, dann wird sie mit dem Wert eins versehen. Wenn nicht, so erhält sie den Wert null. Die Tabelle 5-4 fasst jegliche Parameterwerte aller Klassen zusammen.

Klassen / **Parameter**	**Gebäudegrundriss**	**Ackerfläche**	**Wald**	**Straßennetz**
Rechtwinkligkeit	1	0	0	0
Umfang < 500 m	1	1	0	0
Umfang > 500 m	0	0	1	1
Fläche < 1000 m²	1	0	0	0
1000 m² < Fläche < 10.000 m²	0	1	0	0
Fläche > 10.000 m²	0	0	1	1
Breite < 5 m	1	0	0	1
Anzahl Innenfläche > 5 m	0	0	0	1

Tabelle 5-4: Parameter-Wert-Liste aller Objektklassen

Die Parameter werden für jede Klasse in eine der SOM zugängliche Form, den Parametervektor, gebracht. Da jede Objektklasse mit acht Parametern versehen ist, hat der Parametervektor die Dimension Acht. Ein Beispiel des Parametervektors für die Klasse Gebäudegrundriss stellt sich mathematisch wie folgt dar:

$$x_G = (1, 1, 0, 1, 0, 0, 1, 0)$$

Die Parametervektoren bilden die Eingabemuster für die SOM, mit dessen Hilfe die Interpretation der Kartenobjekte erfolgen soll. Eine Betrachtung dessen soll im nächsten Kapitel erfolgen.

5.2 Interpretation mit SOM

Die Eingabemuster werden in der Trainingsphase der SOM gelernt. Am Ende der Trainingsphase werden durch Ausgabeneuronen der Ausgabeschicht unterschiedliche Klassen ausgegeben. Nach der Trainingsphase kann die Ausführungsphase gestartet werden. In der Ausführungsphase werden Parametervektoren für jedes Kartenobjekt berechnet und als Eingabevektoren von der SOM gelesen. Ein Eingabevektor erregt ein bestimmtes Neuron in der Ausgabeschicht. Je nach erregtem Neuron lässt sich nun die jeweilige Klasse des Objekts ableiten.

5.2.1 Trainingsphase

In der Trainingsphase werden zunächst die Gewichtungen für Ein- und Ausgabeschicht initialisiert. Die Initialisierung der Gewichtung wird durch Pseudozufallszahlen realisiert. Anschließend werden die Parametervektoren in die Eingabeschicht eingegeben. Danach wird die Euklidische Distanz der Parametervektoren zu den Gewichtsvektoren der Ausgabeschicht berechnet. Aus dem Neuron mit der kleinsten Euklidischen Distanz ergibt sich das Erregungszentrum.

Nach der Groborientierung wird die Gewichtung des Erregungszentrums sowie die seiner umliegenden Neuronen in jedem Lernschritt verfeinert (vgl. 2.2.6.4). Das Lernen soll, um das beste Ergebnis zu erzielen, mindestens 200-mal wiederholt werden. Die Funktionen der Lernrate und der Nachbarschaft werden für angestrebte Optimierung der Gewichtung wie folgt festgelegt:

- *Die Lernrate* wird durch die exponentielle Funktion berechnet:

$$\eta(t) = \eta_{max} * exp^{-\frac{t}{t_{max}}}$$

t: aktuelle Lernschritte

t_{max}*: gesamte Lernschritte*

- Die Nachbarschaft wird durch Gaußsche Glockenfunktion (vgl. 2.2.6.3) ermittelt. Wobei sich der Nachbarschaftsradius durch die exponentielle Funktion berechnen lässt:

$$r(t) = r_{max} * exp^{-\frac{t}{t_{max}}}$$

Zur Bestimmung der passenden Größe der SOM wird der Quantisierungsfehler und der topographische Fehler genutzt. Beide hängen von der Anzahl der Objektklassen sowie der Anzahl der Eingabeparameter ab. Quantisierungs- und topographische Fehler der unterschiedlichen SOM-Größen werden in der Tabelle 5-5 aufgelistet. Die Fehler sind durchschnittliche Werte mehrerer Versuche. Aus der Ta-

belle ist abzulesen, dass es ab der SOM Größe 10 x 10 zu niedrigeren Quantisierungs- und topographischen Fehlern kommt. Die Größen 20 x 20 und 30 x 30 sind mit sehr kleinen Fehlern verbunden. Aus diesem Grund wird für die Interpretation die SOM Größe 30 x 30 festgelegt. Die unterschiedliche Darstellung der SOM mit verschiedenen Größen ist im Anhang F zu sehen.

SOM Größe	Quantisierungsfehler	Topographischer Fehler
2 x 2	0,5736	0,5
3 x 3	0,126	0,125
4 x 4	0,078	0,125
5 x 5	0,0065	0
10 x 10	$5{,}776 * 10^{-7}$	0
15 x 15	$1{,}164 * 10^{-8}$	0
20 x 20	$5{,}127 * 10^{-10}$	0
30 x 30	$1{,}577 * 10^{-12}$	0

Tabelle 5-5: Fehler der SOM für ländliche Fläche

5.2.2 Ausführungsphase

Nachdem die SOM trainiert wurde, können in der Ausführungsphase die Objekte interpretiert werden. Die Parameter für jedes Kartenobjekt werden analysiert und bilden einen neuen Parametervektor. Ist dies geschehen, wird dieser neue Parametervektor in die SOM eingegeben. Die Ähnlichkeit des Parametervektors zu den in der Lernphase trainierten Neuronen wird berechnet und das Objekt wird der Klasse des Neurons mit der größten Ähnlichkeit zugeordnet. Die daraus erkannte Objektklasse wird im Resultat ausgegeben. Zur Veranschaulichung der Ergebnisse werden die Objekte mit entsprechender Farbe versetzt und auf der Karte dargestellt (Abbildung 5-4). Die Abbildung 5-3 zeigt die Legende der Klassen mit Farben.

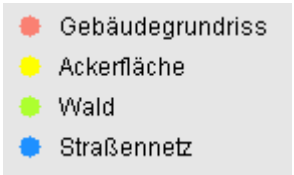


Abbildung 5-3: Legende

Abbildung 5-4: Testkarte mit Ergebnis

Die Ergebnisse können als valide angesehen werden. Vier aus insgesamt 176 Objekten werden falsch erkannt. Die Erkennungsquote liegt bei 97,72%. Die Ackerflächen *A*, *B*, *C* und *D* werden fälschlicherweise als Gebäudegrundrisse erkannt. Dem liegt zugrunde, dass ihre Flächen geringer als 1000 m² sind, was dem Bereich für Gebäudegrundrisse (siehe Tabelle 5-4) entspricht. Der Flächen-Parameter für Ackerfläche dagegen wird für den Bereich von 1000 m² bis 10.000 m² definiert. Eine Unterscheidung der Klassen Gebäudegrundriss und Ackerfläche durch den Umfang macht eine Elimination des Fehlers möglich. Bisher wird sowohl der Umfang der Klasse Gebäudegrundriss als auch der der Ackerfläche zum Bereich kleiner als 500 m eingeordnet. Nun wird dies geändert: Der Umfang der Klasse Gebäudegrundriss wird kleiner als 100 m, der Umfang der Klasse Ackerfläche wird auf 100 m bis 500 m angepasst. Die neue Parameter-Wert-Liste ist in der Tabelle 5-6 einzusehen.

Klassen / Parameter	Gebäudegrundriss	Ackerfläche	Wald	Straßennetz
Rechtwinkligkeit	1	0	0	0
Umfang < 100 m	1	0	0	0
100 m < Umfang < 500 m	0	1	0	0
Umfang > 500 m	0	0	1	1
Fläche < 1000 m²	1	0	0	0
1000 m² < Fläche < 10.000 m²	0	1	0	0
Fläche > 10.000 m²	0	0	1	1
Bereite < 5 m	1	0	0	1
Anzahl Innenfläche > 5 m	0	0	0	1

Tabelle 5-6: Parameter-Wert-Liste aller Objektklassen mit feingranuliertem Umfang

Nun wird die Testkarte mit den optimierten Eingabeparametern erneut interpretiert. In dem neuen Ergebnis ist der Fehler eliminiert (Abbildung 5-5).

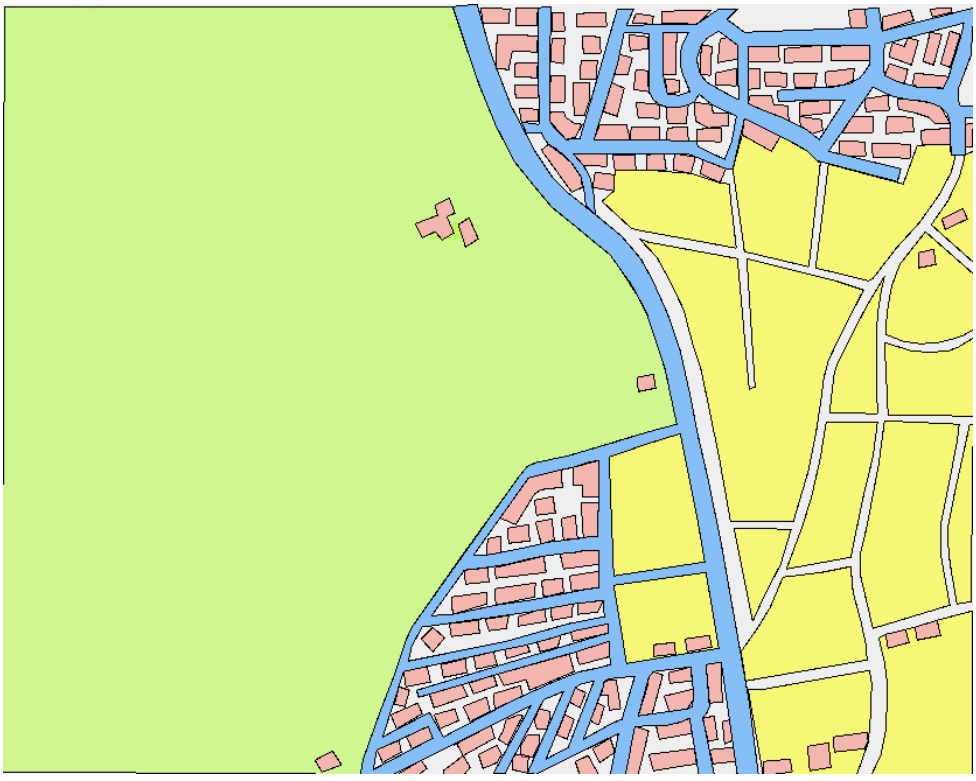


Abbildung 5-5: Optimiertes Ergebnis mit Umfang

Die Optimierung macht deutlich, dass die präzise Auswahl der Parameter zur möglichst fehlerlosen Interpretation eine wesentliche Rolle spielt. Je genauer die Parameter aufgeteilt werden, desto exakter ist das Ergebnis.

5.3 Test mit weiterem Beispiel

Um zu prüfen, wie das Verfahren auf anderen Datensätzen funktioniert, wird eine weitere Testkarte interpretiert. Sie enthält ebenfalls die Objektklassen Gebäudegrundriss, Ackerfläche, Wald und Straßennetz. Die Abbildung 5-6 zeigt die Testkarte, die einen Ausschnitt von Schönaich/Böblingen darstellt. Sie wird ebenfalls auf den gleichen Stuttgarter Stadtplan der Landeshauptstadt Stuttgart digitalisiert und als Shapefile gespeichert.

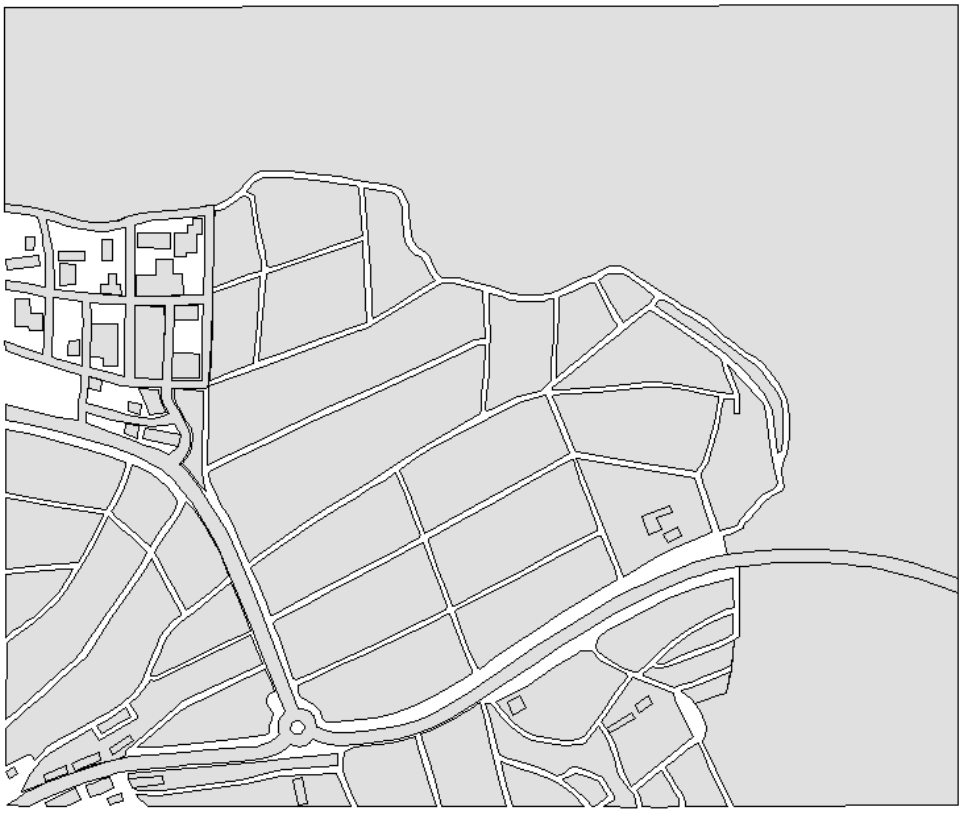


Abbildung 5-6: Zweite Testkarte (Ausschnitt aus Schönaich)

Da die Objektklassen unverändert bleiben, werden die gleichen Merkmale betrachtet und die SOM arbeitet mit den identischen Eingabemustern. Die Parametervektoren für die Kartenobjekte werden ausgewertet und in die SOM eingegeben. Die SOM erzielt in der Ausführungsphase aus den Parametervektoren die Ergebnisse. Sie werden in der Abbildung 5-7 dargestellt. Alle Objekte, abgesehen von zwei Ackerflächen *A* und *B*, die als Gebäudegrundrisse erkannt werden, werden richtig erkannt. Die

Ackerflächen werden dabei fehlerhaft erkannt, weil ihre Fläche sowie ihr Umfang sich im Wertebereich für Gebäudegrundrisse befinden. Die Fehlinterpretation lässt sich nicht vermeiden, da die Objekte sich allein durch die definierten geometrischen Merkmale nicht unterscheiden.

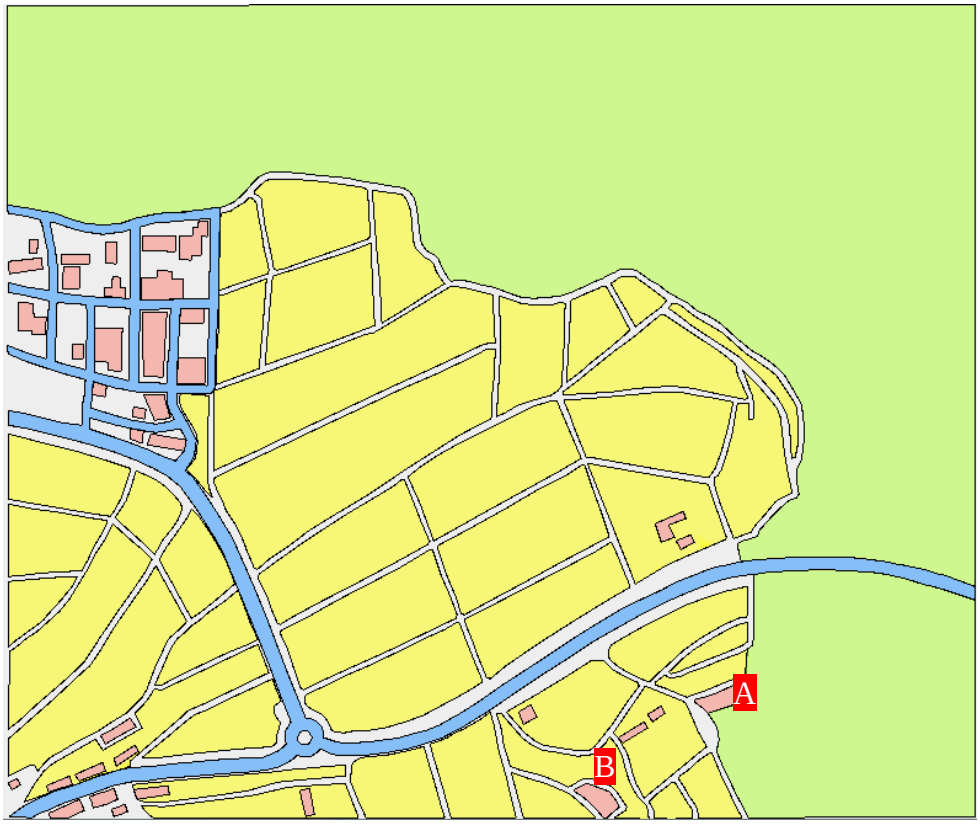


Abbildung 5-7: Zweite Testkarte mit Ergebnis

Das in der zweiten Testkarte erzielte Ergebnis verdeutlicht die Validität des Verfahrens im Hinblick auf Nutzung von weiteren Karten. Die definierten geometrischen Parameter zeigen sich auch für die zweite Testkarte als gültig.

5.4 Qualitätsbetrachtung

Die SOM ist in der Lage, Objekte zu erkennen, auch wenn ihre Eingabeparameter nicht exakt mit einem gelernten Muster übereinstimmen. Es ist sinnvoll, dabei zu untersuchen, inwiefern die Eingabeparameter mit den Musterparametern übereinstimmen. Zu diesem Zweck wird der Begriff *Qualität* definiert. Je größer die Parametervektoren vom Musterparameter abweichen, desto wahrscheinlicher wird das Objekt falsch erkannt.

Die Berechnung der *Qualität* erfolgt aus dem Ähnlichkeitsvergleich zwischen dem Parametervektor eines Objektes und seinem entsprechenden Muster. Ein Zähler wird mit null initialisiert. Der Zähler zählt, wie oft der Parameter des Objekts gleich dem Parameter des Musters ist. Zum Schluss wird die

prozentuale Größe aus der Division zwischen dem Zähler und der Größe des Parametervektors als die *Qualität* des Objekts bezeichnet:

Zähler = 1;

Wenn (Objekt.Parameter_1 ist gleich Muster.Parameter_1) dann

Zähler++;

Wenn (Objekt.Parameter_2 ist gleich Muster.Parameter_2) dann

Zähler++;

Wenn (Objekt.Parameter_3 ist gleich Muster.Parameter_3) dann

Zähler++;

Wenn (Objekt.Parameter_4 ist gleich Muster.Parameter_4) dann

Zähler++;

...

Wenn (Objekt.Parameter_N ist gleich Muster.Parameter_N) dann

Zähler++;

Ende;

*Qualität = Zähler / N * 100%;*

Die Testkarte aus Abbildung 5-8 wird hinsichtlich ihrer Qualität untersucht. Die Objekte hundertprozentiger *Qualität* werden in Abbildung 5-8 dargestellt. Es ist ersichtlich, dass die Objekte ohne hundertprozentiger Qualität ausgefiltert werden, auch wenn sie mittels SOM richtig erkannt werden können (siehe Abbildung 5-5). Das jedoch zeigt, dass die SOM die Erkennung richtig durchführen kann, ohne dass die Objekte den gelernten Mustern exakt entsprechen müssen.

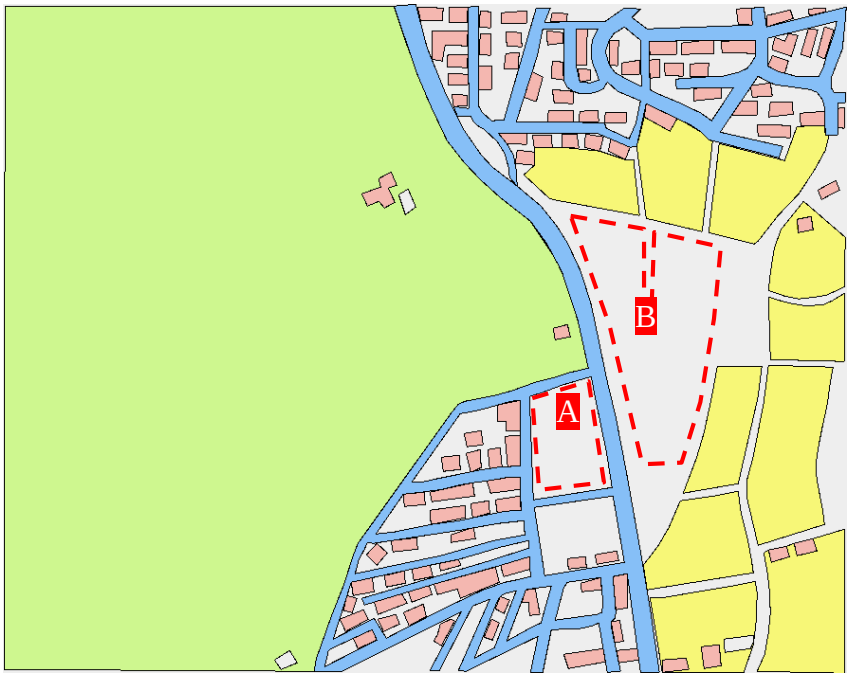


Abbildung 5-9: Objekte mit Qualität = 100%

Es soll nun näher betrachtet werden, warum die Objekte die hier ohne hundertprozentige *Qualität* sind, in der SOM erkannt werden können. Beispielsweise hat das Objekt *A* in der Abbildung 5-9 den Eingabevektor:

$$E_A = (1, 0, 1, 0, 0, 1, 0, 0, 0)$$

welcher dem Muster:

$$E_{Ackerfläche} = (0, 0, 1, 0, 0, 1, 0, 0, 0)$$

nicht hundertprozentig entspricht, da die Ackerfläche aus rechten Winkeln besteht, während Ackerflächen nicht rechtwinklig sein sollen.

Das Objekt *B* in der Abbildung 5-9 besitzt den Eingabevektor:

$$E_B = (0, 0, 0, 1, 0, 1, 0, 0, 0)$$

welcher nicht hundertprozentig dem Muster:

$$E_{Ackerfläche} = (0, 0, 1, 0, 0, 1, 0, 0, 0)$$

entspricht, da die Länge der Ackerfläche *B* mit 541 m länger als 500 m ist. Der Parameter *Umfang <500 m* wird nicht erfüllt. Wird der Schwellenwert für die Länge auf beispielsweise 600 m erhöht, so wird die Ackerfläche *B* mit hundertprozentiger *Qualität* erkannt. Daraus folgt, dass die Bestimmung

der Schwellenwerte eine wichtige Rolle spielt. Mit der Änderung des Schwellenwerts kann das Ergebnis der Interpretation variieren. Wird der Schwellenwert zu hoch angesetzt, wird ein Objekt mit einem Wert darunter möglicherweise nicht richtig erkannt. Umgekehrt wird ein Objekt mit dem Wert über dem Schwellenwert eventuell falsch erkannt, wenn der Schwellenwert zu niedrig angesetzt wird. Es soll mit mehreren Kandidaten des Schwellenwerts experimentiert werden, um den idealen Schwellenwert zu ermitteln. Um das manuelle Vergleichen der Ergebnisse zu automatisieren, kann die *Qualität* eingesetzt werden. Der Wert mit den meisten mit hundertprozentiger *Qualität* erkannten Objekten wird als geeigneter Schwellenwert bestimmt.

5.5 Diskussion

Die Untersuchung zeigt, dass die einzelnen Kartenobjekte ihrer entsprechenden Klasse zugeordnet werden können. Der entscheidende Punkt dabei ist das Vorhandensein der jeweiligen geometrischen Eigenschaften der einzelnen Klassen, so dass sie sich voneinander unterscheiden lassen. Für neue Objektklassen müssen neue Merkmale erkannt und definiert werden. Wird z.B. die Klasse Freifläche in die Testkarten einbezogen, können die Klassen Freifläche und Ackerfläche anhand der bereits angewendeten Merkmale nicht differenziert werden. Die Freifläche würde als Ackerfläche fehlinterpretiert. Beide verfügen über keine trennenden Merkmale in ihrer Struktur und variieren nicht deutlich in ihren geometrischen Größen. Für neue Merkmale kann die Topologie der Objekte hinzugezogen werden. Nachbarschaft von Ackerflächen oder das Umschließen von Gebäudegrundrissen durch Freifläche sind hier Beispiele. Eine interpretatorische Trennung von Klassen wie Park und See ist schwer umzusetzen, da beide eine eher willkürliche Geometrie und Topologie in sich tragen, was die Trennung problematisch macht.

Die Berechnung der geometrischen Werte wie Größe, Länge, etc. basiert in der vorliegenden Arbeit auf den Testkarten, die lediglich Ausschnitte von Orten abbilden. Die Klassen Gebäudegrundriss sowie Ackerfläche können dank ihrer kleinen geometrischen Ausbreitung in den Ausschnitten vollständig dargestellt werden. Das Straßennetz sowie der Wald dehnen sich über den Ausschnitt hinaus aus. Für den Fall einer Betrachtung über die gewählten Ausschnitte hinaus ist eine erneute Berechnung der Geometrie sinnvoll. Für die Untersuchung der vorliegenden Arbeit jedoch, sind die Ausschnitte ausreichend, da sie der Unterscheidung der Klassen bereits genügen. Zur weiteren Untersuchung ist der Einsatz eines Fuzzy-Algorithmus für die Festlegung der unscharfen Schwellenwerte denkbar. Das unscharfe Entscheidungsverfahren erlaubt die Verarbeitung von Wertrelativierungen.

Die qualitativen Eigenschaften, etwa dass Gebäudegrundrisse über rechte Winkel oder dass Straßennetze über Innenflächen verfügen, sind robuster als die quantitativen Merkmale, wie die Fläche oder die Länge. Die Schwellenwerte werden experimentell festgelegt, da teilweise Überlappungen der Wertbereiche verschiedener Klassen vorliegen. Die Ergebnisse variieren mit der Änderung der Schwellenwerte. Die Festlegung der qualitativen Merkmale dagegen ist eindeutiger.

Das Ergebnis kann optimiert werden, wenn möglichst viele Merkmale einkalkuliert werden. Zu diesem Zweck können komplexere Strukturen als weitere Merkmale dienen.Wenn ein Objekt eine komplexere Struktur enthält, kann geprüft werden, ob die Struktur auf ein bestimmtes Objekt hinweisen kann. Beispielsweise kann ein Straßennetz identifiziert werden, wenn ein Autobahnkreuz oder ein Kreisverkehr vorhanden sind. Weitere Merkmalen können die Anzahl der Kanten, der Linienverlauf, etc. sein. Da aber bereits die definierten Merkmale zu guten Ergebnissen führen, werden solche Alternativmerkmale hier nicht untersucht.

6 Interpretation des Kartentyps

Nachdem die Interpretation der Kartenobjekte vorgestellt wurde, soll nun die Interpretation der Kartentypen erläutert werden.

Die Datensätze für die Interpretation des Kartentyps sind die durch die im Kapitel 4 beschriebene Suche erworbenen Shapefiles. In einem Shapefile können jeweils nur Punkte, Linien oder Polygone enthalten sein. Shapefiles mit punktförmigen Objekten werden in der Untersuchung ignoriert, da die Punkte wegen den ähnlichen geometrischen Eigenschaften in der Karteninterpretation wenig Bedeutung haben. Es werden ausschließlich Karten mit linienförmigen und polygonförmigen Objekten in der Interpretation berücksichtigt. Wie im Kapitel 4.3 erläutert, werden 10018 Dateien aus dem Internet im Dateisystem gespeichert. Sie werden nach dem Typ der Elemente geprüft. Die Dateien aus Punkten werden ausgefiltert und der Rest wird in 3028 Dateien mit linienförmigen Objekten und 5134 Dateien mit polygonförmigen Objekten getrennt.

Karten können nach verschiedenen Merkmalen der Kartengrafik und des damit verbundenen Maßstabs in unterschiedliche Kartentypen differenziert werden [Hake et al. 2002]. Auf die in Shapefile gespeicherten Geometrien und Topologien wird zugegriffen und sie werden analysiert, um Muster des Kartentyps für die SOM festzulegen. Anders als bei der Interpretation der einzelnen Objekte, wird die Topologie bzw. die Nachbarschaft zwischen den Objekten hier berücksichtigt. Durch die geometrische sowie topologische Charakterisierung können Muster für die Kartentypen erstellt werden. Nachdem die Muster der Kartentypen für die SOM in der Trainingsphase gelernt wurden, wird ein Parametervektor für jede Karte berechnet. Die Parametervektoren der Karten werden in die SOM eingegeben und können nach ihrem Kartentyp erkannt werden.

Im Folgenden werden die Interpretationen der Karten sowohl mit linienförmigen als auch mit polygonförmigen Objekten erläutert.

6.1 Karten mit linienförmigen Objekten

Karten mit linienförmigen Objekten können in unterschiedliche Typen aufgeteilt werden, wie beispielsweise Höhenlinienkarten, Flusskarten und Straßenkarten. Damit die Kartentypen automatisch erkannt werden können, müssen die Karten durch menschliche Betrachter interpretiert werden können. Nicht alle Karten können durch menschliche Betrachter erkannt werden. Dies liegt an zwei Gründen. Zum einen befinden sich zu wenig Linien in den Karten, zum anderen beinhalten die Karten zu wenig eindeutige Charakteristika. Aus diesem Grund werden 250 Shapefiles für die Arbeit ausgewählt, die eine ausreichende Charakterisierung für die Interpretation enthalten. Die Karten lassen sich in unterschiedliche Typen untergliedern (Abbildung 6-1):

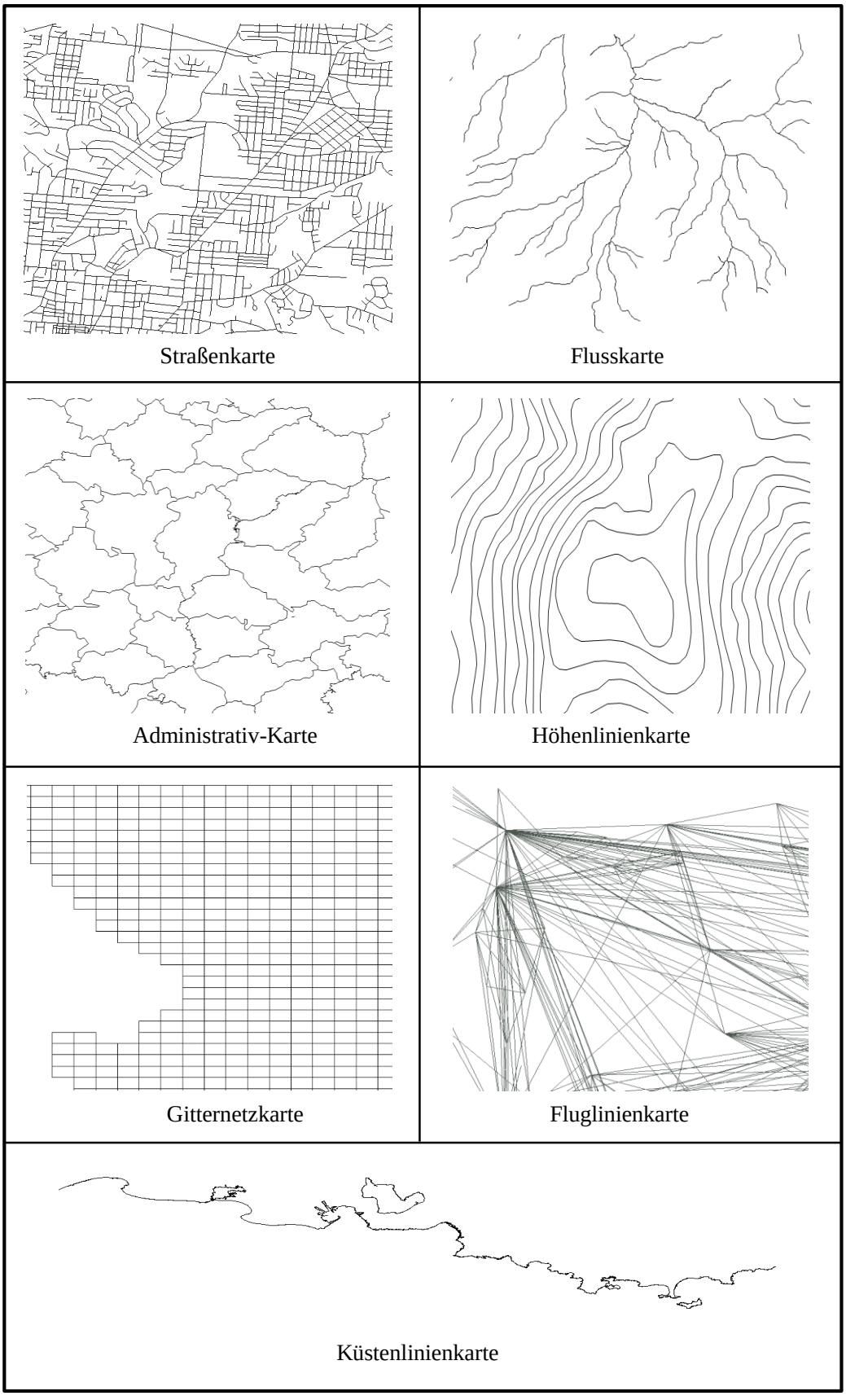


Abbildung 6-1: Kartentypen mit linienförmigen Objekten

- Straßenkarte
- Flusskarte
- Administrativ-Karte
- Küstenlinienkarte
- Höhenlinienkarte
- Gitternetzkarte
- Fluglinienkarte

Die Auswahl der Kartentypen folgt dem Kriterium „Unterschiedlichkeit“ der typischen geometrischen sowie topologischen Eigenschaften der Karten. Als Beispiel soll hier die Höhenlinienkarte erläutert werden. Höhenlinien haben einen bestimmten festen Abstand in der Vertikalen. Auf einem Punkt können keine zwei Höhenlinien gleichzeitig erscheinen, d.h. es gibt keinen Schnittpunkt auf einer Höhenlinienkarte. Somit lassen sich die Höhenlinienkarten von anderen Karten unterscheiden. Im Folgenden werden die Merkmale der ausgewählten Kartentypen im Detail diskutiert.

6.1.1 Merkmalsdefinition

6.1.1.1 Knotentypen

Ein wesentliches Merkmal der Linien sind die s.g. Knoten, die den Anfang und das Ende einer Linie bzw. eines Linienabschnittes bezeichnen. Die Knoten können in verschiedene Typen differenziert werden. Die Abbildung 6-2 zeigt eine Liste von Knotentypen aus [Sester 1995].

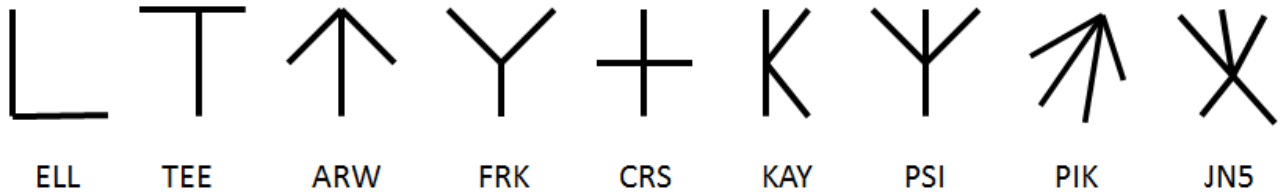


Abbildung 6-2: Knotentypen (nach [Sester 1995])

Nun werden die Kartentypen hinsichtlich ihrer jeweiligen Knotentypen analysiert.

1. Straßenkarte

Prinzipiell können alle Knotentypen in Straßenkarten vorkommen. Sie beschreiben vielfältige Möglichkeiten, wie die Straßen im Straßennetz in einem Punkt zusammentreffen. Die Abbildung 6-3 stellt beispielhaft fünf Knotentypen in einem Straßennetz dar.

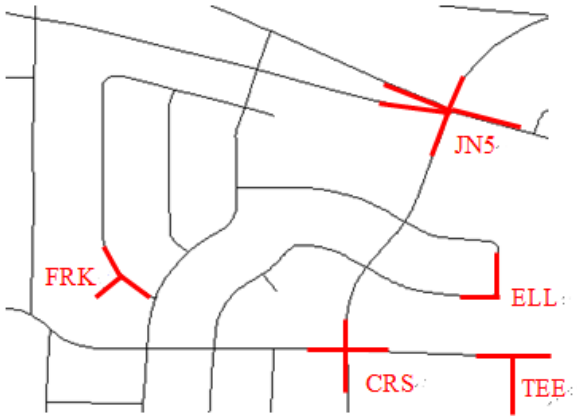


Abbildung 6-3: Beispiele der Knotentypen im Straßennetz

2. Flusskarte

Flüsse besitzen die spezifische Charakterisierung, dass sie sich ihrem Lauf nach in mehrere Äste verzweigen. Die Knoten an den Abzweigungen können als FRK-Knoten bezeichnet werden. Es handelt sich in diesem Fall jedoch um eine veränderte Version vom FRK-Knoten mit Grad 2. Der Fluss wird am Knoten nicht in zwei Zweige aufgesplittet, sondern der Lauf des Flusses bleibt unverändert und ein zusätzlicher Zweig entwickelt sich aus dem Fluss. Der FRK-Knoten für einen Fluss erfüllt drei Eigenschaften:

- Zwei Linien schneiden sich beim FRK-Knoten, der Knotengrad wird als zwei bezeichnet
- Es wird kein rechter Winkel um FRK-Knoten gebildet
- Eine Linie endet bzw. beginnt am FRK-Knoten.

Die Abbildung 6-4 zeigt, dass eine große Anzahl an FRK-Knoten auf einer Flusskarte zu finden sind.

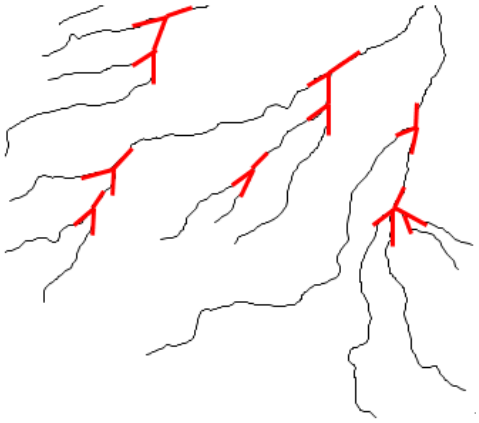


Abbildung 6-4: Flusskarte mit mehreren FRK-Knoten (Grad 2)

Die Häufigkeit der FRK-Knoten wird als ein Merkmal festgelegt, um Flusskarten von anderen Kartentypen zu unterscheiden. Straßenkarten enthalten ebenfalls FRK-Knoten, allerdings in geringerer Anzahl. Da die FRK-Knoten hauptsächlich in Straßen- und Flusskarten erscheinen, werden die Häufigkeiten aus einer Stichprobe für die beiden Kartentypen berechnet. Die Häufigkeit kann durch die prozentuale Proportion der Anzahl der FRK-Knoten zu der Anzahl sämtlicher Kanten formuliert werden. Es wird nicht mit der Anzahl der gesamten Knoten verglichen, da Flusskarten angesichts der ihnen eigenen Darstellung mehr Knoten enthalten als Straßenkarten.

Die folgende Abbildung 6-5 zeigt die prozentuale Häufigkeit der FRK-Knoten von jeweils fünf Fluss- und Straßenkarten. Die blaue Linie repräsentiert die Häufigkeit der FRK-Knoten in Straßenkarten. Sie liegt unter 10%. Die rote Linie zeigt die Häufigkeit der FRK-Knoten in Flusskarten. Der kleinste Wert befindet sich bei 10%. Der Schwellenwert wird allerdings als 30% festgelegt, um das Merkmal für Flusskarten hervorzuheben. Flüsse mit der Häufigkeit weniger als 30% werden nicht berücksichtigt.

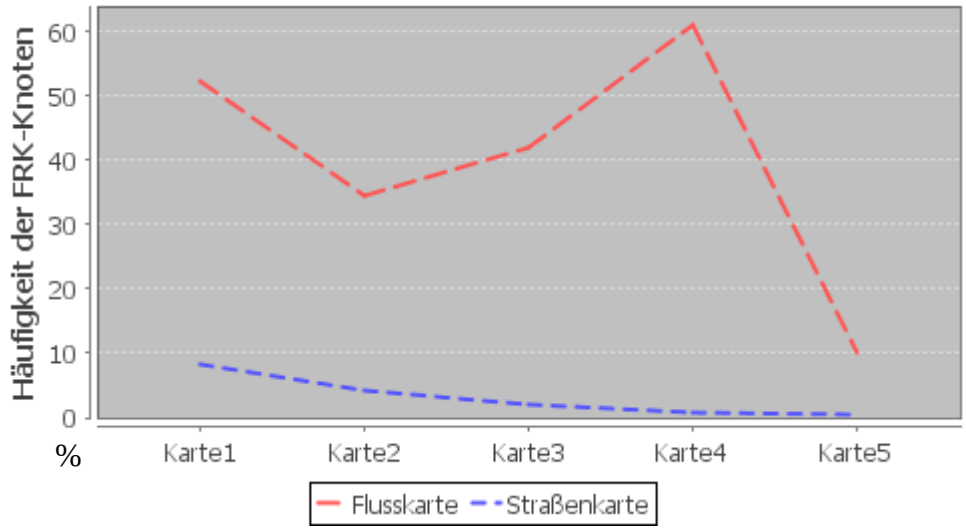


Abbildung 6-5: Häufigkeit der FRK-Knoten (Grad 2)

3. Administrativ-Karte

Die Administrativ-Karte stellt die innere administrative Gliederung der Staaten, Länder bzw. Regierungsbezirke dar. Durch ihre netzförmige Topologie sind eine Reihe der FRK-Knoten mit Grad 3 auf der Karte enthalten (Abbildung 6-6).

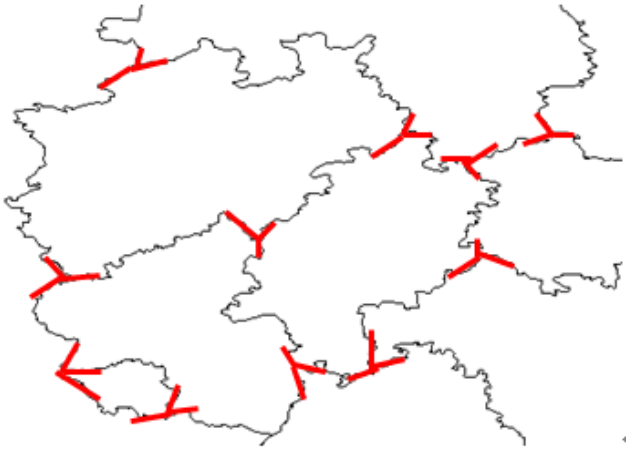


Abbildung 6-6: Administrativ-Karte mit mehreren FRK-Knoten (Grad 3)

Im Unterschied zum FRK-Knoten der Flusskarten berühren sich drei Linien am FRK-Knoten der Administrativ-Karten, d.h. drei Linien enden bzw. beginnen am FRK-Knoten. Der Knotengrad wird infolgedessen als drei bezeichnet. Das Merkmal für Administrativ-Karten wird folgendermaßen definiert: Über 50% der Linien auf der Karte besitzen sowohl am Anfangspunkt als auch am Endpunkt FRK-Knoten.

4. Küstenlinien- und Höhenlinienkarte

Küstenlinien werden durch Linien entlang der Küste bezeichnet, um das Land vom Meer zu trennen. Aufgrund dieser Definition schneiden sich die Küstenlinien nicht miteinander. Auch bei den Höhenlinien gibt es wegen ihrer Äquidistanz keine Schnittpunkte. So kommen keine Knotentypen in Küstenlinienkarten sowie Höhenlinienkarten vor.

5. Gitternetzkarte

In Gitternetzkarten sind aufgrund ihrer Rechtwinkligkeit die rechtwinkligen Knotentypen zu finden. Dies sind die ELL-, TEE- und CRS-Knoten. Abgesehen von den Gitternetzkarten erscheinen diese Knotentypen hauptsächlich in Straßenkarten. Die Häufigkeit der Knoten wird zur Abgrenzung von Gitternetzkarten zu Straßenkarten ermittelt.

Um die relative Häufigkeit der rechten Winkel auszudrücken, wird der Prozentanteil der Anzahl der rechten Winkel zu der gesamten Knotenanzahl berechnet. Da die rechten Winkel bei der Erstellung der Karte nicht sicher mit 90° vektorisiert werden, werden auch Winkel mit 5%-iger Abweichung toleriert.

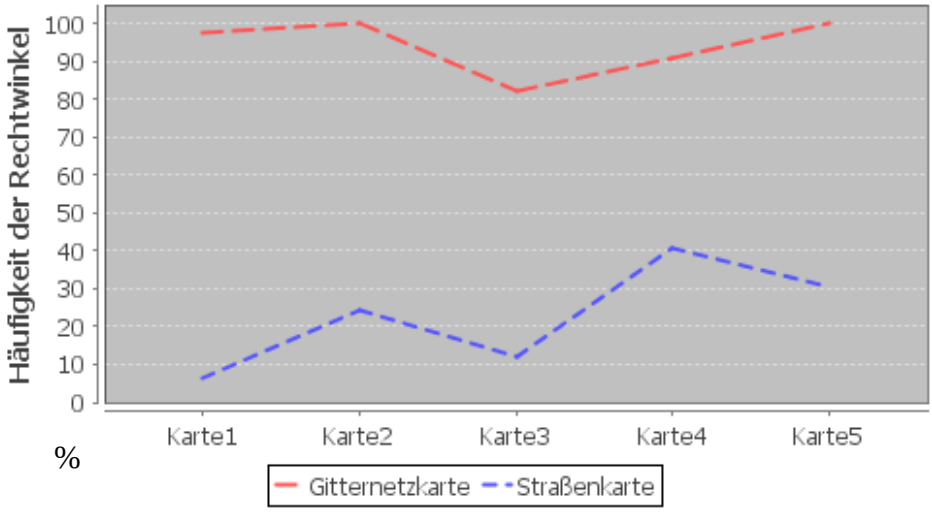


Abbildung 6-7: Häufigkeit der rechten Winkel

In der Abbildung 6-7 werden die Häufigkeiten der rechten Winkel jeweils für fünf Gitternetz- und Straßenkarten dargestellt. Die Häufigkeit für Gitternetzkarte liegt über 50%, die für Straßenkarten im Bereich 5% bis 50%.

6. Fluglinienkarte

Karten mit Fluglinien bestehen aus einer großen Menge einander kreuzender Geraden, so dass überwiegend Knotentypen mit hohem Grad, beispielsweise JN5, JN6, JN7 [Anders 2007] oder darüber, vorkommen. Diese erkennbare Eigenheit der Fluglinien auf den Fluglinienkarten trennt sie von den anderen Kartentypen. Der JN10-Knoten wird in dieser Arbeit definiert und zu dem Zweck der Trennung angewandt. JN10-Knoten entstehen, wenn zehn Linien sich an einem Knoten miteinander schneiden.

Neben den behandelten Knotentypen können weitere Merkmale Kartentypen charakterisieren. Sie werden im Folgenden erläutert.

6.1.1.2 Nicht-schneidende Linien

Wie bereits erwähnt wurde, kreuzen sich sowohl die Küsten- als auch die Höhenlinien nicht untereinander. Diese spezifische Eigenschaft kann die Küstenlinien- und Höhenlinienkarte von den anderen Karten abheben. Sie liegt vor, wenn kein gemeinsamer Schnittpunkt zwischen zwei Linien ermittelt werden kann.

6.1.1.3 Geradlinigkeit

Gerade Linien sind ausschließlich in Fluglinien- und Gitternetzkarten enthalten. Dagegen zeigen sich kurvige Linien in den anderen Kartentypen. Die Geradlinigkeit kann interpretiert werden, wenn eine

Linie lediglich aus zwei Punkten besteht oder wenn alle Winkel zwischen den Linienabschnitte 180° sind.

6.1.1.4 Geschachtelte geschlossene Linien

Höhenlinien sind verhältnismäßig geschlossene konzentrisch ineinander geschachtelte Linien (Abbildung 6-8). Höhenlinien stellen eine Erhebung dar, indem mehrere Höhenlinien ineinander geschachtelt und geschlossen sind und die Höhe zur Mitte zunimmt. Umgekehrt wird eine Mulde gezeigt, wenn mehrere Höhenlinien ineinander geschachtelt und geschlossen sind und die Höhe zur Mitte abnimmt.

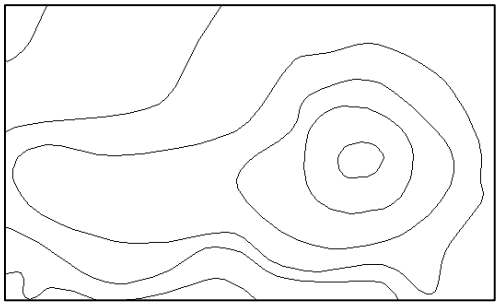


Abbildung 6-8: Geschachtelte geschlossene Linien in Höhenlinienkarten

In anderen Kartentypen können alleinstehende geschlossene Linien vorkommen. So wird beispielsweise ein Kreisverkehr als geschlossene Linie in Straßenkarten dargestellt. Inseln auf einer Küstenlinienkarte werden ebenfalls mithilfe einer geschlossenen Linie abgebildet. Um also die Höhenlinienkarten von anderen Karten abzugrenzen, ist es notwendige Bedingung, dass mindestens drei geschlossene Linien ineinander geschachtelt vorliegen. Es sollen die kleinsten Koordinaten der inneren Linie, jeweils in vertikaler und horizontaler Richtungen, kleiner als die Koordinaten der jeweils äußeren Linie sein. Außerdem sollen die größten Koordinaten der jeweils äußeren Linie, größer als die Koordinaten der inneren Linie sein.

Aus den diskutierten Merkmalen können nun die Parameter der Kartentypen abgeleitet werden. Alle Parameter sowie die entsprechenden Werte für alle Kartentypen werden in der Tabelle 6-1 zusammengefasst.

Parameter \ Kartentypen	Straßen-karte	Fluss-karte	Administrativ-Karte	Küstenlinien-karte	Höhen-linien-karte	Gitternetz-karte	Fluglinien-karte
JN10-Knoten > 0	0	0	0	0	0	0	1
FRK-Knoten(Grad 2) < 30%	1	0	1	1	1	1	1
FRK-Knoten(Grad 2) > 30%	0	1	0	0	0	0	0
FRK-Knoten(Grad 3) < 50%	1	1	0	1	1	1	1
FRK-Knoten(Grad 3) > 50%	0	0	1	0	0	0	0
Rechter Winkel < 5%	0	1	1	1	1	0	1
5% < Rechter Winkel < 50%	1	0	0	0	0	0	0
Rechter Winkel > 50%	0	0	0	0	0	1	0
Nicht-schneidende Linie	0	0	0	1	1	0	0
Geradlinigkeit	0	0	0	0	0	1	1
Geschachtelte geschlossene Linien	0	0	0	0	1	0	0

Tabelle 6-1: Parameter-Wert-Liste aller Kartentypen mit linienförmigen Objekten

6.1.2 Interpretation mit SOM

Nachdem die Parametervektoren der Kartentypen festgelegt wurden, können sie als Eingabemuster in der SOM trainiert werden. Die im Kapitel 5.2 beschriebene SOM kann hier beibehalten werden. Nur für die Interpretation unbekannter Karten sollen die Parametervektoren der Karten ermittelt werden.

Vor der Ermittlung der Parametervektoren muss der Fall berücksichtigt werden, dass bei der Digitalisierung eine Linie aus mehreren Linien zusammengeführt wurde (siehe Abbildung 6-9). Das ergibt sich aus einer Unterbrechung des Arbeitsschritts der manuellen Digitalisierung. Die Konsequenz daraus ist, dass dadurch sich schneidende Linien dargestellt sind, wo sie eigentlich die Eigenschaft nicht-schneidende Linie haben. Dazu wird geprüft, ob die Linien *A* und *B* sich am Schnittpunkt *K* berühren. Die gebrochenen Linien sollen zusammengefügt werden.

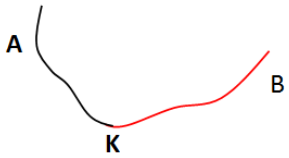


Abbildung 6-9: Zusammenführende Linien

Nun werden die Karten in der Ausführungsphase der SOM hinsichtlich ihres Kartentyps erkannt. Die Ergebnisse sind in der Tabelle 6-2 aufgelistet.

Kartentypen / Anzahl	Straßen-karte	Fluss-karte	Administrativ-Karte	Küsten-linienkarte	Höhen-linienkarte	Gitternetz-karte	Fluglinien-karte
Karten zum Test	73	70	50	25	15	16	1
Karten richtig erkannt	58	59	43	25	15	16	1

Tabelle 6-2: Ergebnisse für Kartentypen mit linienförmigen Objekten

250 Karten wurden getestet. 86,8% der Karten werden richtig erkannt. Im Anhang G.1 werden einige Beispiele zu den richtig bzw. falsch interpretierten Karten dargestellt. Die Interpretationsquote hängt davon ab, wie viele der ausgewählten Merkmale die Testkarten beinhalten. Eine Flusskarte wird fälschlicherweise als Straßenkarte interpretiert. Die Kurven dieser Flusskarte werden stets mit kleinen rechten Winkeln digitalisiert, so dass zahlreiche rechte Winkel erkannt werden und die Karte so irrtümlich als Straßenkarte interpretiert wird. Der Ausschnitt in der Abbildung 6-10 zeigt einen Ausschnitt der Flusslinie.

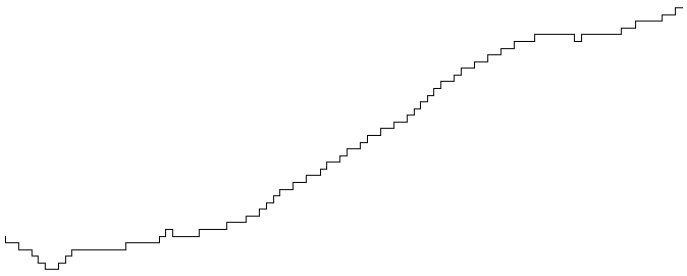


Abbildung 6-10: Flusslinie mit kleinen Rechtecken

6.2 Karten mit polygonförmigen Objekten

In diesem Abschnitt werden Karten mit Objekten polygonaler Darstellungsform betrachtet. Kartenobjekte wie Gebäude und Regionen werden großenteils flächig dargestellt. Ähnlich den Karten mit linienförmigen Objekten müssen die zu interpretierenden Karten mit polygonförmigen Objekten durch menschliche Betrachter analysiert werden können. In der manuellen Vorarbeit werden für die Untersuchung 250 Shapefiles ausgewählt. Diese Karten bestehen aus sechs verschiedenen Kartentypen (siehe Abbildung 6-11):

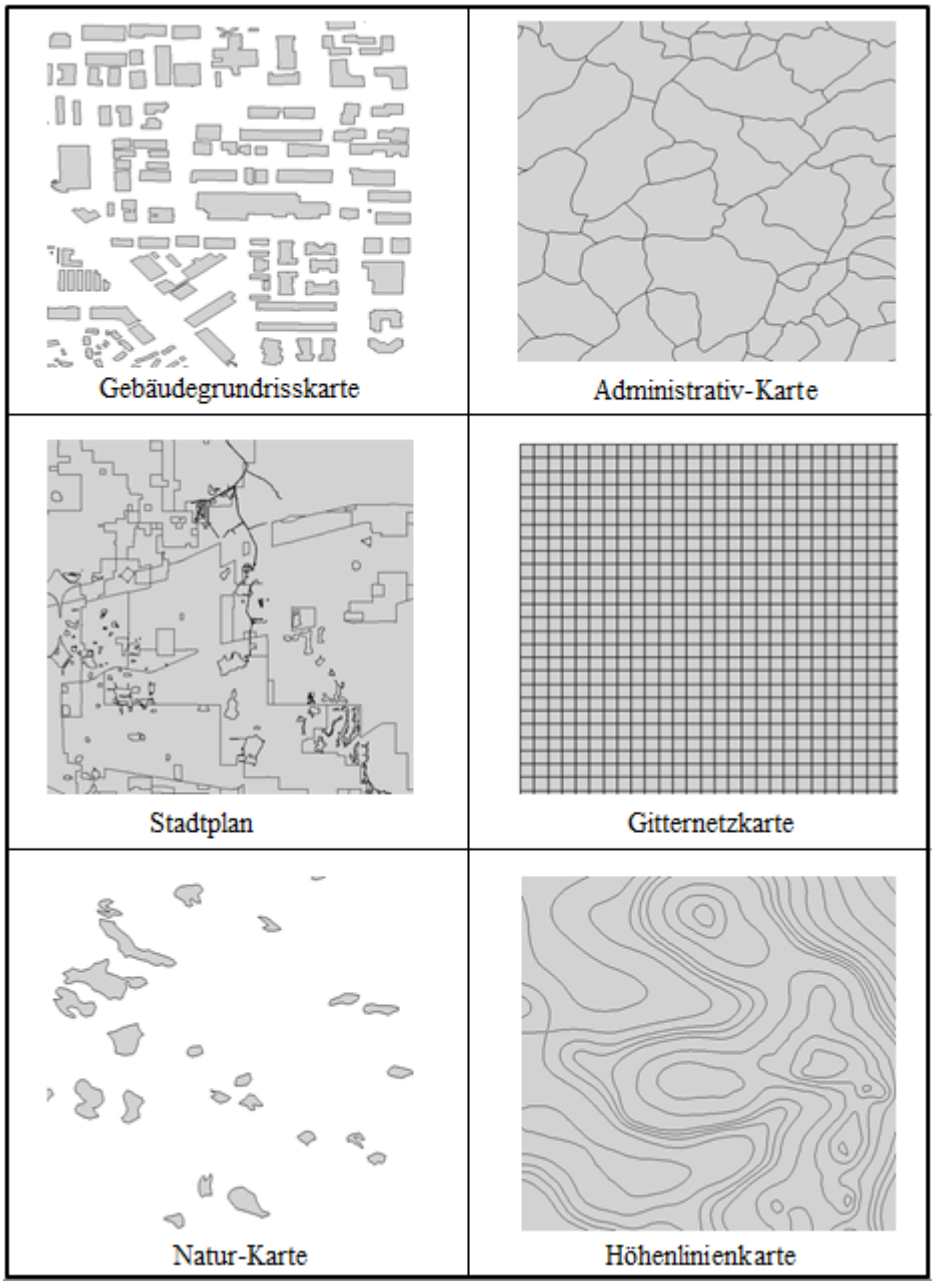


Abbildung 6-11: Kartentypen mit polygonförmigen Objekten

- Gebäudegrundrisskarte
- Administrativ-Karte
- Stadtplan
- Gitternetzkarte
- Natur-Karte
- Höhenlinienkarte

In die Natur-Karten werden Karten für See, Teich, Becken, Park sowie Wald einbezogen.

Die Kartentypen mit polygonförmigen Objekten sollen ebenfalls durch ihre jeweils eigenen geometrischen sowie topologischen Merkmale voneinander unterschieden werden.

6.2.1 Merkmalsdefinition

In diesem Abschnitt werden die Merkmale der polygonförmigen Objekte für die unterschiedlichen Kartentypen analysiert.

6.2.1.1 Gleichmäßige Flächen

Die Kartenobjekte einer Karte können verschiedene Flächenmaße besitzen (Abbildung 6-12 links). Gitternetzkarten jedoch enthalten Polygone mit gleichem Flächenmaß (Abbildung 6-12 rechts).

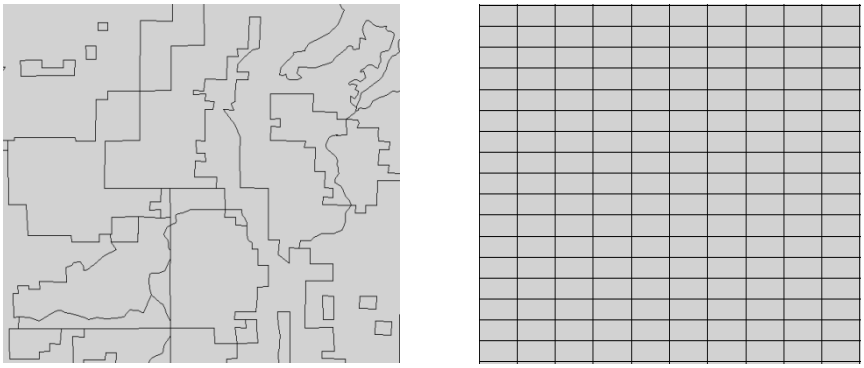


Abbildung 6-12: Karten mit ungleichmäßigen (links) und gleichmäßigen (rechts) Flächen

Das Gleichmaß der Fläche wird wie folgt geprüft: Die Differenz der Fläche A eines Polygons zu der durchschnittlichen Fläche aller Polygone A_m wird berechnet. Aufgrund verschiedener Koordinatensysteme ist es nicht angebracht, die Differenzen als absolute Werte zu verwenden. So soll der Prozentanteil der Differenzen zum Durchschnitt berechnet werden:

$$\frac{A - A_m}{A_m} \times 100\%$$

Um die Gitternetzkarte von anderen Kartentypen unterscheiden zu können, werden Prozentanteile von jeweils fünf Gitternetz- und fünf anderen Karten berechnet und in der Abbildung 6-13 dargestellt. Die blaue Linie zeigt die Werte der Karten mit ungleichmäßigen Flächen. Sie liegt unter dem Wert 10%. Die rote Linie drückt das Gleichmaß in den Gitternetzkarten aus. Die Prozentanteile sind über 10%.

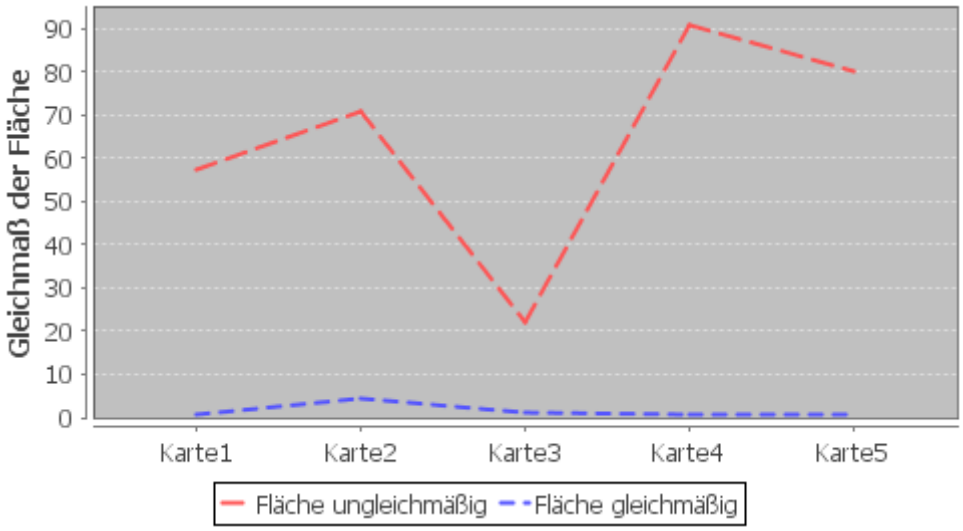


Abbildung 6-13: Gleichmaß der Flächen

6.2.1.2 Rechter Winkel

Die Rechtwinkligkeit spielt, wie bei den Karten mit linienförmigen Objekten, bei der Interpretation der Karten mit polygonförmigen Objekten eine wichtige Rolle. Es werden zwei Fälle differenziert:

- Rechter Winkel häufiger als 50%. Die Gebäudegrundriss- und Gitternetzkarten werden erkennbar durch rechte Winkel typisiert. Der Anteil der rechten Winkels liegt über 50%.
- Ohne rechten Winkel. Die Karten mit Höhenlinien, Administrationsgrenzen sowie Seen sind im Normalfall ohne rechte Winkel ausgestattet.

6.2.1.3 Nachbarschaft

Nun sollen die Nachbarschaften der Polygone analysiert werden. In der vorliegenden Arbeit hat Nachbarschaft die Bedeutung, dass jedes Polygon mit mindestens einem anderen Polygon benachbart ist (siehe Abbildung 6-14 links). Wenn lediglich ein Teil der Polygone Nachbarn haben, wie in der Abbildung 6-14 rechts dargestellt, wird die Karte nicht mit dem Merkmal *benachbart* gekennzeichnet.

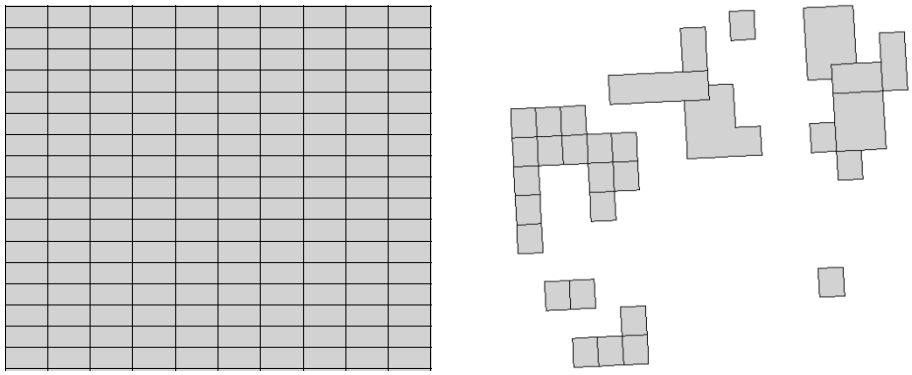


Abbildung 6-14: Alle Polygone sind benachbart (links) und nicht alle Polygone sind benachbart (rechts)

Um eine Nachbarschaft zu ermitteln, wird auf s.g. Knotennachbarn oder Kantennachbarn geprüft [Esri 2013]. Knotennachbarn liegen dann vor, wenn die Kanten verschiedener Polygone sich an einem Punkt überkreuzen oder berühren. Kantennachbarn sind vorhanden, wenn Polygone gemeinsame oder sich berührende Kanten besitzen. Die überlappenden Nachbarn werden für die Shapefiles in dieser Arbeit nicht geprüft, da sie in den Daten selten auftreten.

6.2.1.4 Umschlossene Polygone

Eine Struktur, in der mehrere Objekte innerhalb eines Objektes liegen, erscheint immer wieder in Stadtplänen oder in Höhenlinienkarten. Ein Stadtplan enthält Gebäude, Verkehrsflächen, Freiflächen etc.. Diese werden jeweils von einer Gebietsgrenze umschlossen. In einer Höhenlinienkarte werden die Polygone immer kleiner bzw. größer, sobald sich die Höhe ändert. Diese Charakterisierung kann Stadtpläne und Höhenlinienkarten von anderen Karten differenzieren.

6.2.1.5 Umschlossene Gebäude

Um Stadtpläne von Höhenlinienkarten trennen zu können, werden die umschlossenen Objekte analysiert. Der Stadtplan kann anhand der Identifizierung eines Gebäudes, das umschlossen ist, erkannt werden. Da die inneren Höhenlinien im Normalfall kurvig sind, ist es zur Vereinfachung der Erkennung der Gebäude ausreichend, wenn umschlossene vierseitige Polygone mit vier rechten Winkeln identifiziert werden können. Der rechte Winkel wird durch 90° mit einem Puffer von 5° definiert.

6.2.1.6 Geschachtelte Polygone

Zu einer weiteren Trennung zwischen Stadtplan und Höhenlinienkarte kann die geschachtelte Struktur beitragen. Wie bereits erwähnt wurde, sind die Höhenlinien ineinander geschachtelt.

Nun werden für die Kartentypen Parameter aus den diskutierten Merkmalen abgeleitet. Alle Parameter sowie die entsprechenden Werte werden in der Tabelle 6-3 zusammengefasst.

Kartentypen / Parameter	Gebäude-grundrisskarte	Administrativ-karte	Stadtplan	Natur-karte	Gitternetz-karte	Höhenlinien-karte
Fläche gleichmäßig	0	0	0	0	1	0
Rechter Winkel > 50%	1	0	0	0	1	0
Kein rechter Winkel	0	0	0	1	0	1
Benachbart	0	1	0	0	1	0
Umschlossene Polygone	0	0	1	0	0	1
Umschlossene Gebäude	0	0	1	0	0	0
Geschachtelte Polygone	0	0	0	0	0	1

Tabelle 6-3: Parameter-Wert-Liste für Kartentypen mit polygonförmigen Objekten

6.2.2 Interpretation mit SOM

Die im Kapitel 5.2 beschriebene SOM ist auch hier gültig. Die Parametervektoren der Kartentypen mit polygonförmigen Objekten werden in der Trainingsphase von der SOM gelernt. Danach werden die 250 vorher ausgewählten Karten interpretiert und ihren Kartentypen zugewiesen. Die Ergebnisse sind in Tabelle 6-4 aufgelistet.

Kartentypen / Anzahl	Gebäude-grundrisskarte	Administrativ-Karte	Stadtplan	Natur-Karte	Gitternetz-karte	Höhenlinien-karte
Karten zum Test	69	89	8	54	29	1
Karten richtig erkannt	58	69	8	54	29	1

Tabelle 6-4: Ergebnisse für Kartentypen mit polygonförmigen Objekten

87,6% der Karten werden richtig erkannt. Im Anhang G.2 werden einige Beispiele zu den richtig bzw. falsch interpretierten Karten dargestellt. Der Interpretationserfolg hängt direkt von der Anzahl der definierten Merkmale in den Testkarten ab. Beispielsweise wird die Gebäudegrundrisskarte fälschlicherweise als Natur-Karte interpretiert, da sie Häuser beliebiger Formen enthält (Abbildung 6-15).

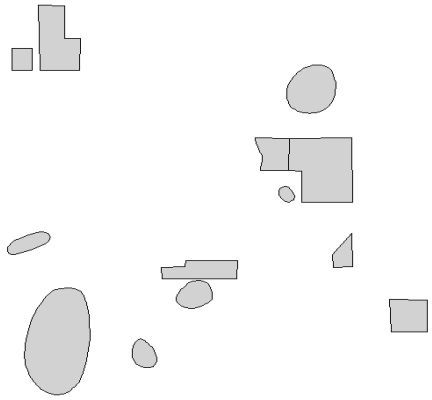


Abbildung 6-15: Gebäudegrundrisse in beliebigen Formen

6.3 Verwendung von Zusatzinformationen

Zusatzinformationen sind Informationen, die nicht eigens zum Inhalt der Karte gehören, die aber trotzdem der Interpretation zuträglich sind. Die Webseite oder der Dateiname der Karte können eine solche Zusatzinformation sein.

6.3.1 Verwendung des Dateinamens

Der Dateiname einer Karte enthält mitunter treffende Stichwörter, die den Karteninhalt zusammenfassen. Solche Stichwörter können den Kartentyp demnach identifizieren helfen. In der Tabelle 6-5 sind Beispiele der im Dateiname vorkommenden Stichwörter und die daraus erkannten Kartentypen aufgelistet. Auch die Anzahl der Shapefiles im Dateisystem ist einzusehen. Es wird eine große Menge von nützlichen Stichwörtern im Dateinamen entdeckt. Trotzdem sind die Dateinamen nicht immer zuverlässig. Beispielsweise enthält der Name einer Karte das Wort „city“, die Karte ist jedoch kein Stadtplan. Es wird lediglich die Stadtgrenze dargestellt. In einem anderen Beispiel treten zwei Stichwörter „river“ und „highway“ gleichzeitig in einem Dateiname auf, so dass die Datei zwei Kartentypen zugeordnet werden kann.

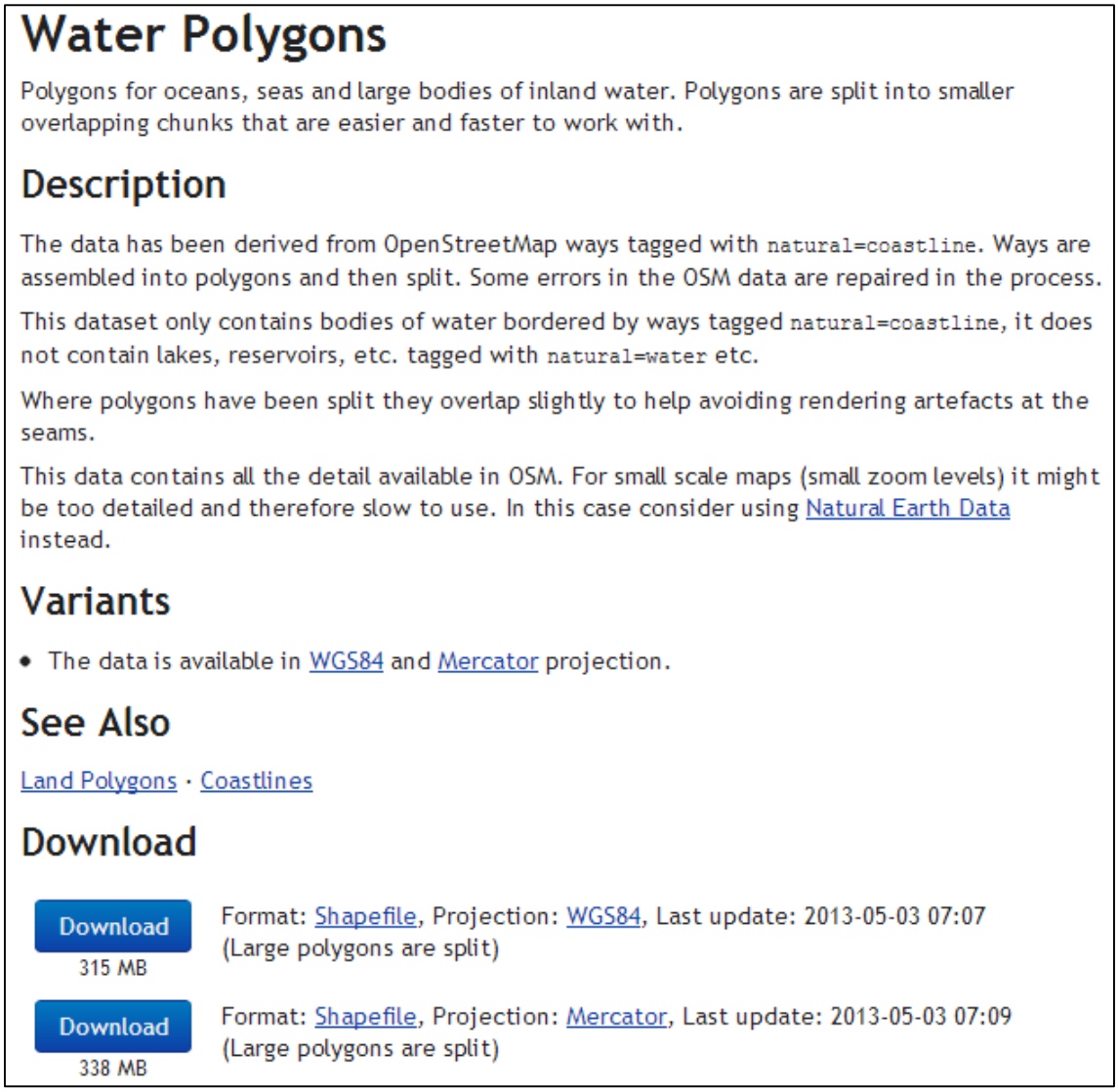
In einigen Fällen können Stichwörter im Dateinamen die Prüfung der Plausibilität der Ergebnisse für die Karteninterpretation ermöglichen. Wenn der durch ein Stichwort angedeutete Kartentyp dem erkannten Ergebnis entspricht, dann ist die Interpretation mutmaßlich korrekt, andernfalls ist das Ergebnis potenziell fehlerhaft.

Kartentypen	Treffendes Stichwort im Dateiname	Anzahl der Karten mit Stichwort	Anzahl der gesamten Karten
Natur-Karte	Lake, natural, water, basin, park, forest	6	54
Gebäudegrundrisskarte	Building, house	3	49
Stadtplan	City, urban, town	2	8
Administrativ-Karte	County, administrative, region, district, state	34	99
Flusskarte	River, stream	9	40
Straßenkarte	Highway, road, street	23	43

Tabelle 6-5: Dateiname der Karten mit Stichwort

6.3.2 Verwendung der Webseite

Wie im Kapitel 4 erläutert wurde, werden die Shapefiles aus dem Internet heruntergeladen. Die Shapefiles werden auf ihrer jeweiligen Webseite, auf der sie verlinkt sind, zum Herunterladen angeboten. Auf dieser Webseite steht im Rahmen des Möglichen eine Zusammenfassung oder Erläuterung über das Shapefile. Die Abbildung 6-16 gibt so ein Beispiel. Die Beschreibung auf der Webseite deutet darauf hin, dass es sich hier um eine polygonförmige Gewässerkarte handelt. Diese Information kann zur Beurteilung der Ergebnisse durch Karteninterpretation nützlich sein. Es ist jedoch nicht immer möglich eine Information dieser Art der Webseite zu entnehmen. Die Abbildung 6-17 zeigt ein Beispiel mit einer Liste von Shapefiles einer Webseite. Auf dieser Webseite können mehrere Shapefils heruntergeladen werden. Es befindet sich keine Beschreibung auf der Webseite.



Water Polygons

Polygons for oceans, seas and large bodies of inland water. Polygons are split into smaller overlapping chunks that are easier and faster to work with.

Description

The data has been derived from OpenStreetMap ways tagged with `natural=coastline`. Ways are assembled into polygons and then split. Some errors in the OSM data are repaired in the process.

This dataset only contains bodies of water bordered by ways tagged `natural=coastline`, it does not contain lakes, reservoirs, etc. tagged with `natural=water` etc.

Where polygons have been split they overlap slightly to help avoiding rendering artefacts at the seams.

This data contains all the detail available in OSM. For small scale maps (small zoom levels) it might be too detailed and therefore slow to use. In this case consider using Natural Earth Data instead.

Variants

- The data is available in WGS84 and Mercator projection.

See Also

Land Polygons · Coastlines

Download

Download 315 MB — Format: Shapefile, Projection: WGS84, Last update: 2013-05-03 07:07 (Large polygons are split)

Download 338 MB — Format: Shapefile, Projection: Mercator, Last update: 2013-05-03 07:09 (Large polygons are split)

Abbildung 6-16: Webseite mit Beschreibung des Shapefiles (aus http://openstreetmapdata.com/data/water-polygons)

Name	Last modified	Size	Description
Parent Directory		-	
2012-09-12-06-59.garmin-gmapsupp.zip	12-Sep-2012 09:00	4.1M	
2012-09-12-06-59.garmin-img.zip	12-Sep-2012 09:00	4.1M	
2012-09-12-06-59.garmin-mapsource-installer.exe	12-Sep-2012 09:00	3.8M	
2012-09-12-06-59.osm.bz2	12-Sep-2012 09:01	7.1M	
2012-09-12-06-59.shp.zip	12-Sep-2012 09:01	5.7M	
2012-09-12-06-59.txt	12-Sep-2012 09:01	1.0K	
2012-09-12-07-00.garmin-gmapsupp.zip	03-May-2013 09:07	4.1M	
2012-09-12-07-00.garmin-img.zip	03-May-2013 09:07	4.1M	
2012-09-12-07-00.garmin-mapsource-installer.exe	03-May-2013 09:07	3.8M	
2012-09-12-07-00.osm.bz2	03-May-2013 09:07	7.1M	
2012-09-12-07-00.shp.zip	03-May-2013 09:07	5.7M	
2012-09-12-07-00.txt	03-May-2013 09:07	1.0K	
latest.garmin-gmapsupp.zip	03-May-2013 09:07	4.1M	
latest.garmin-img.zip	03-May-2013 09:07	4.1M	
latest.garmin-mapsource-installer.exe	03-May-2013 09:07	3.8M	
latest.osm.bz2	03-May-2013 09:07	7.1M	
latest.shp.zip	03-May-2013 09:07	5.7M	
latest.txt	03-May-2013 09:07	1.0K	

Abbildung 6-17: Webseite mit Auflistung der Shapefiles (aus http://labs.geofabrik.de/haiti/)

6.3.2.1 Text Mining

Um wichtige Informationen über Karten von deren Webseiten erfassen zu können, wird das Verfahren Text Mining angewandt. Text-Mining-Verfahren dienen dazu, Informationen aus unstrukturierten Textdaten zu interpretieren. Text Mining wird als die maschinelle Entdeckung von zuvor unbekanntem Wissen in Textdokumenten verstanden [Felden 2006a]. Mit Text-Mining-Verfahren können Maschinen Textfragmente automatisch miteinander verknüpfen und Informationen verwertbar machen, so dass die Benutzer das nützliche Wissen aus den Dokumenten erschließen können.

Das Text-Mining gehört zu einer Kombination der Gebiete Computerlinguistik und Statistik. Prinzipiell werden die Methoden des Data Mining, das strukturierte Daten aus Datenbanken analysiert, auf unstrukturierte Daten bzw. Texte angewandt. Hierfür werden Techniken aus verschiedenen wissenschaftlichen Disziplinen angewandt, zu denen das Data Mining, das Information Retrieval, die Computerlinguistik, die Statistik sowie Intelligente Software Agenten gehören [Felden 2006b].

6.3.2.2 TFIDF

Es gibt mittlerweile viele gängige Tools und Verfahren zum Text Mining. Das Text Mining wird in der vorliegenden Arbeit zur Prüfung der Vorkommenshäufigkeit der Stichwörter angewandt, um so die Kartentypen aus den Webseiten zu erkennen. Der Algorithmus TFIDF (Term Frequency, Inverse

Document Frequency) wird für die automatische Schlüsselworterkennung verwendet. Nachdem Stoppwörter eliminiert wurden, analysiert TFIDF wie häufig ein Wort in einem Dokument vorkommt. Stoppwörter sind Wörter, die sehr häufig auftreten und keine Relevanz für den Dokumentinhalt besitzen, etwa Artikel, Konjunktionen und Präpositionen.

Drei wichtige Definitionen für den TFIDF sind:

- Termfrequenz (tf). Sie kennzeichnet die Häufigkeit eines Wortes innerhalb eines Dokuments und deutet darauf hin, dass die Wörter wichtig sind, weil sie oft in einem Dokument erwähnt werden.

$$tf = \frac{N_i}{\sum N_k}$$

N_i: Anzahl eines Wortes im Dokument

$\sum N_k$: Anzahl aller Wörter im Dokument

- Dokumentenfrequenz (df) bezeichnet die Anzahl der Dokumente, in denen das Wort auftritt.
- Inverse Dokumentenfrequenz (idf) ist nützlich für das Herausfinden signifikanter Wörter, die in möglichst wenigen Dokumenten erscheinen. Dafür wird die Worthäufigkeit in der ganzen Dokumentsammlung berechnet.

$$idf_t = log\frac{N}{df_t}$$

N: Anzahl der Dokumente

df_t: Anzahl der Dokumente, die die Wörter enthalten

In der vorliegenden Arbeit geht es jeweils um eine Webseite für eine Karte. Insofern wird keine Dokumentsammlung betrachtet. So wird lediglich der Begriff Termfrequenz für diese Arbeit eingesetzt. Die Dokumentenfrequenz sowie die Inverse Dokumentenfrequenz werden ignoriert. Die Schlüsselwörter werden nach den Kartentypen in Gruppen aufgeteilt:

- Gruppe Gebäudegrundriss: *building, house*
- Gruppe Stadt: *city, urban, town*
- Gruppe Region: *region, district, state, county*
- Gruppe Fluss: *river, stream*
- Gruppe Straße : *highway, street, road*

Um einen Kartentyp zu identifizieren, ist es wichtig, dass lediglich eine Gruppe der Schlüsselwörter auf der Webseite erscheint. Mit mehreren Gruppen auf einer Webseite kann der Kartentyp nicht eindeutig erkannt werden.

Um die Webseiten der Shapefiles auszuwerten, werden die Webseiten von 5000 Shapefiles heruntergeladen. Die Dokumentenfrequenz für die Webseiten wird berechnet. Es können 105 Shapefiles dem zugehörigen Kartentyp zugeordnet werden. Eine Quote von 2,1%.

6.4 Diskussion

In diesem Kapitel wird die Interpretation der Kartentypen untersucht. Die Untersuchung zeigt, dass die SOM geeignet für die Interpretation des Kartentyps ist. Das Verfahren basiert auf der Analyse der Merkmale der Vektorkarten. Die Auswahl der Merkmale folgt dem Kriterium, dass sie die Kartentypen voneinander trennen können. Es ist problematisch, Merkmale für die Natur-Karten zusammenzufassen, da die Objekte auf der Karte unregelmäßig verteilt sind und keine spezifische geometrische Form besitzen. Die Natur-Karten können von den anderen Kartentypen unterschieden werden, indem sie die Eigenschaften der anderen Karten nicht enthalten. Im Unterschied zur Interpretation der Kartenobjekte, werden die Merkmale hier nicht mit absoluten Werten wie z.B. die Länge der Linie bezeichnet, da der Maßstab verschiedener Karten unterschiedlich ist. Ein Vergleich der Karten ist nicht angebracht.

Die Interpretation wird genauer, je mehr Merkmale einkalkuliert werden können. Bereits erforschte Verfahren zur Datenstruktur können weitere Merkmale liefern. Beispielsweise werden einige charakteristische Strukturen für Straßennetz in [Heinzle et al. 2007] (vgl. 3.1) untersucht. Solch spezifische Strukturen können die Merkmale der vorliegenden Arbeit ergänzen. Alternativmerkmale werden hier nicht untersucht, da die definierten Merkmale bereits zu guten Ergebnissen führen.

Die Breite der erkannten Karten wird durch die strikte Auswahl der Merkmale eingeschränkt. Beispielsweise kann eine Karte nicht als Flusskarte erkannt werden, wenn die Anzahl der FRK-Knoten (Grad 2) einer Flusskarte unter dem Bereich des determinierten Merkmals liegt. Auch die Administrativ-Karte wird nicht erkannt, wenn Regionen auf einer Administrativ-Karte nicht benachbart sind.

Eine Vorbearbeitung der Daten ist daher erforderlich. Es werden Karten ausgewählt, die durch die bloßen Augen interpretiert werden können. Falls die Karten ohne spezifische Charakterisierung nicht vorher zu filtern sind, werden sie ebenfalls in die SOM einbezogen. Dabei werden sie fälschlicherweise einem Kartentyp zugeordnet, dem sie nicht angehören. Die Ergebnisse der Interpretation hängen davon ab, wie viel der definierten Merkmale die Testkarten besitzen. Je mehr Merkmale die Testkarten aufweisen, desto höher ist die Interpretationsquote. Zur Automatisierung der Vorbearbeitung kann die im Kapitel 5.4 erklärte Qualität genutzt werden. Karten ab gewisser Qualität zeigen sich als plausibel interpretiert.

Wenn eine Karte eines hier nicht definierten Kartentyps ins Verfahren eingeschlossen wird, kann der Kartentyp nicht richtig interpretiert werden. Für die Interpretation neuer Kartentypen müssen entsprechende Merkmale festgelegt werden. Beispielsweise werden Flurstückskarten in diese Arbeit nicht miteinbezogen. Die Abbildung 6-18 zeigt das Beispiel einer Flurstückskarte. Zur Interpretation der Flurstückskarte kann die Parallelität der Flurstücke als Merkmal definiert werden. Außerdem soll die Flurstückskarte vom Stadtplan unterschieden werden, da die Flurstückskarte über eine Reihe von Ähnlichkeiten mit dem Stadtplan verfügt. Um dies zu erreichen, müssen die Merkmale des Stadtplans konkretisiert werden. So enthält der Stadtplan z.B. ein Straßennetz, die Flurstückkarte dagegen nicht.

Abbildung 6-18: Beispiel Flurstückskarte

7 Interpretation des Maßstabs

Nachdem bereits die Interpretationen der Kartenobjekte und Kartentypen behandelt wurden, wird in diesem Kapitel auf Ansätze zur Interpretation des Maßstabs eingegangen.

Der Maßstab ist das Verhältnis einer Kartenstrecke zu der entsprechenden Strecke in der Natur [Hake et al. 2002]. Karten unterschiedlicher Maßstäbe beinhalten unterschiedliche Darstellungen. Je größer der Maßstab, desto genauer sind die Darstellungen auf der Karte. Bestimmte Strukturen werden ausschließlich in größeren Maßstäben angezeigt. Gebäudegrundrisse werden beispielsweise nicht auf Karten kleineren Maßstabs angezeigt. Auch wird eine Straße auf einer Karte mit großem Maßstab als Polygon, auf einer Karte mit kleinem Maßstab als Linie dargestellt. Die Maßstäbe werden in folgende Maßstabsbereiche gegliedert [Hake et al. 2002]:

- Große Maßstäbe: 1:10.000 und größer,
- Mittlere Maßstäbe: kleiner als 1:10.000 bis etwa 1:300.000,
- Kleine Maßstäbe: kleiner als 1:300.000.

Eine digitale Karte besitzt keinen festen Maßstab im herkömmlichen Sinne, da dieser je nach Zoomstufe flexibel ist. In der vorliegenden Arbeit soll der Erfassungsmaßstab interpretiert werden. Er ist gleich dem Maßstab der ursprünglichen Karte, aus der die digitale Karte vektorisiert wurde.

Zur Interpretation des Maßstabs ist es denkbar, das Verhältnis der Kartenausdehnung zum Maßstab zu untersuchen. Je größer die Ausdehnung, desto kleiner der Maßstab. Eine Weltkarte hat einen kleineren Maßstab als ein Stadtplan. Dies ist jedoch nicht allgemeingültig, denn Karten mit gleicher Ausdehnung können durchaus verschiedene Maßstäbe besitzen. So orientiert sich die automatische Interpretation an den übrigen für den Maßstab verwertbaren Eigenschaften und Informationen von digitalen Karten. Diese sind die Mehrfachrepräsentation und der Detaillierungsgrad.

7.1 Interpretation mittels Mehrfachrepräsentation

Mehrfachrepräsentation in GIS bezeichnet die verschiedenartige Darstellung gleicher Realweltobjekte in verschiedenen Karten. Die unterschiedlichen Darstellungen ergeben sich aus den unterschiedlichen thematischen Schwerpunkten der jeweiligen Karten, aus verschiedenen Sichtweisen auf die entsprechenden realen Strukturen und aus dem jeweiligen Maßstab [Sester 2000].

So ergeben sich je nach Erfassungsmaßstab unterschiedlich realitätsgetreue Repräsentationen. Während die Darstellung in Karten größeren Maßstabs die genauere geometrische Struktur widerspiegeln soll, kann sie in Karten kleineren Maßstabs lediglich als symbolischer Stellvertreter dienen. So lässt sich letztlich der Maßstab aus der entsprechenden Repräsentation ableiten. In der Arbeit wird dies anhand eines Beispiels, dem Kreisverkehr erläutert.

In [Balley et al. 2004] wird die Mehrfachrepräsentation eines Kreisverkehrs (Abbildung 7-1) illustriert. Es werden drei Repräsentationen für denselben Kreisverkehr der Realwelt aufgezeigt. Repräsentation 1 ist polygonförmig und wird auf der Karte mit dem Maßstab von 1:5.000 bis 1:50.000 angezeigt. Repräsentation 2 ist punktförmig und wird im kleineren Maßstab von 1:50.000 bis 1:100.000 abgebildet. Repräsentation 3 ist linienförmig und zeigt sich in Navigationskarten des Maßstabs 1:10.000 für die städtische Fläche und 1:50.000 für die ländliche Fläche.

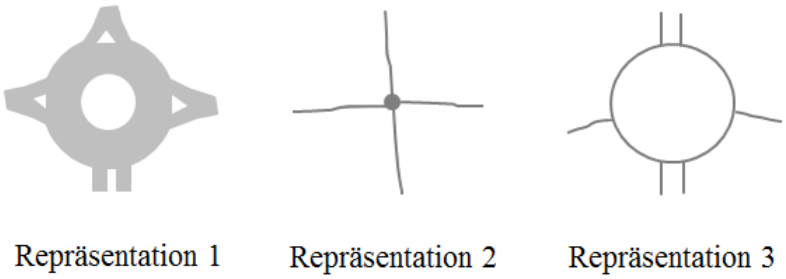


Abbildung 7-1: Alternative geometrische Repräsentationen für denselben Kreisverkehr (nach [Balley et al. 2004])

Sind Repräsentation 1 bzw. Repräsentation 3 auf einer Karte abgebildet, so ist dies ein Hinweis, dass die Karte einen größeren Maßstab besitzt.

Bevor jedoch der Maßstab aus einer Repräsentation abgeleitet werden kann, muss ein Kreisverkehr einer Karte identifiziert werden. In der vorliegenden Arbeit soll es sich um einen Kreisverkehr handeln, der dem der Repräsentation 3 in Gestalt ähnlich ist. Ein solcher Kreisverkehr digitaler Karten soll folgende Eigenschaften erfüllen:

- In die geschlossene Kreis-Linie münden vier bis sechs Linien, die Straßen ein. Diese bestehen aus einzelnen Linien oder aus zwei parallelen Linien, die die Richtungsfahrbahnen darstellen.
- Die geschlossene Linie hat die Fläche 500 – 5000 m² (aus [ADAC 2010] abgeleitet).
- Die geschlossene Linie bildet einen Kreis, und wird wie folgt definiert:

Der Kreis in einem Shapefile besteht aus Linienabschnitten. Da die Genauigkeit von der Digitalisierung abhängt, wird die Anzahl der Linienabschnitte auf mindestens zehn definiert. Der Zentriwinkel μ erfüllt:

$$\mu = \frac{360°}{n}$$

Mit tolerablem Puffer von:

$$\frac{50°}{n}$$

Insgesamt werden 300 Shapefiles mit linienförmigen Objekten aus dem Dateisystem (vgl. 4.3) auf das Vorliegen eines Kreisverkehrs geprüft. In 47 Shapefiles (Kartenbeispiele siehe Anhang E) kann ein Kreisverkehr oder mehrere Kreisverkehre der hier gesuchten Repräsentationsweise identifiziert werden. Das Vorliegen der Kreisverkehre indiziert hier den großen Maßstab. Neben dem Kreisverkehr ist es genauso denkbar, dass Autobahnkreuze, Sportstadien sowie weitere Objekte in ihrer Repräsentationsweise Indikatoren für den jeweils vorliegenden Maßstab sein können.

Die oben diskutierte Mehrfachrepräsentation untersucht Repräsentationen unterschiedlicher Realitätstreue. Die Erscheinungsform war dabei jeweils verschieden. Beim folgenden zweiten Ansatz der Interpretation des Maßstabs, ist die Erscheinungsform auch bei verschiedenen Maßstäben identisch, nur das Maß des Details, der Detaillierungsgrade, der Darstellungen ist unterschiedlich.

7.2 Interpretation mittels Detaillierungsgrad

Bei der Interpretation mittels Detaillierungsgrad lassen sich Objekte der realen Welt auf den digitalen Karten unterschiedlichen Maßstabs mit verschiedenen Detaillierungsgraden erkennen. Auf der Karte größeren Maßstabs werden die Karten detaillierter, auf der Karte kleineren Maßstabs weniger detailliert dargestellt. Dieser Ansatz beabsichtigt daher, aus den verschiedenen Detaillierungsgraden den jeweiligen Maßstab zu ermitteln. Die Detaillierungsgrade werden hinsichtlich der Anzahl der Stützpunkte sowie des Abstands von Linien analysiert.

Der Vergleich der Detaillierungsgrade setzt nötigerweise voraus, dass sich die zu vergleichenden Karten auf den gleichen Kartentyp beziehen. Zwei Karten unterschiedlichen Typs können nicht verglichen werden. So kann beispielsweise eine Gebäudegrundrisskarte nicht mit einer Seekarte verglichen werden.

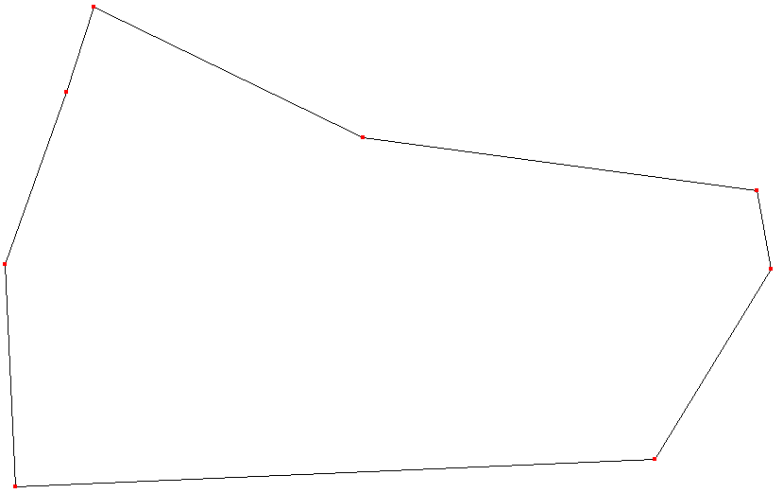
7.2.1 Anzahl der Stützpunkte

Die Polygone und Linien auf einer Vektorkarte werden durch Stützpunkte gebildet. Um die Erscheinungsform eines Objekts auf einer Karte mit großem Maßstab detailgetreu ausdrücken zu können, wird eine Reihe von Stützpunkten benötigt. Ist der Maßstab verhältnismäßig kleiner, so ist die Anzahl der Stützpunkte verringert und somit der Detaillierungsgrad reduziert.

7.2.1.1 Karten auf gemeinsamen Gebiet

In diesem Abschnitt wird die Anzahl der Stützpunkte gleicher Gebiete in Karten unterschiedlichen Maßstabs verglichen. Die Abbildung 7-2 zeigt ein Beispiel der Darstellung der Französischen Süd- und Antarktisgebiete aus zwei Weltkarten (politische Karten) unterschiedlichen Maßstabs. Die Darstellung a) mit dem Maßstab 1:10.000.000 besitzt deutlich mehr Stützpunkte und Objekte als die Darstellung b) mit dem Maßstab 1:110.000.000.

Ausschnitt a) Maßstab: 1:10.000.000



Ausschnitt b) Maßstab: 1:110.000.000

Abbildung 7-2: Detaillierungsgrade für die Französischen Süd- und Antarktisgebiete mit unterschiedlichen Maßstäben

Die Tabelle 7-1 zeigt ein weiteres Beispiel für die Auswirkung der Maßstäbe auf die Anzahl der Stützpunkte. Die drei Weltkarten (politische Karten) sind jeweils mit Maßstäben: 1:10.000.000, 1:50.000.000 und 1:110.000.000 dargestellt. Die Anzahl der Stützpunkte auf der Karte mit dem Maßstab 1:10.000.000 ist fast 80-fach höher als die Anzahl auf der Karte mit dem Maßstab 1:110.000.000.

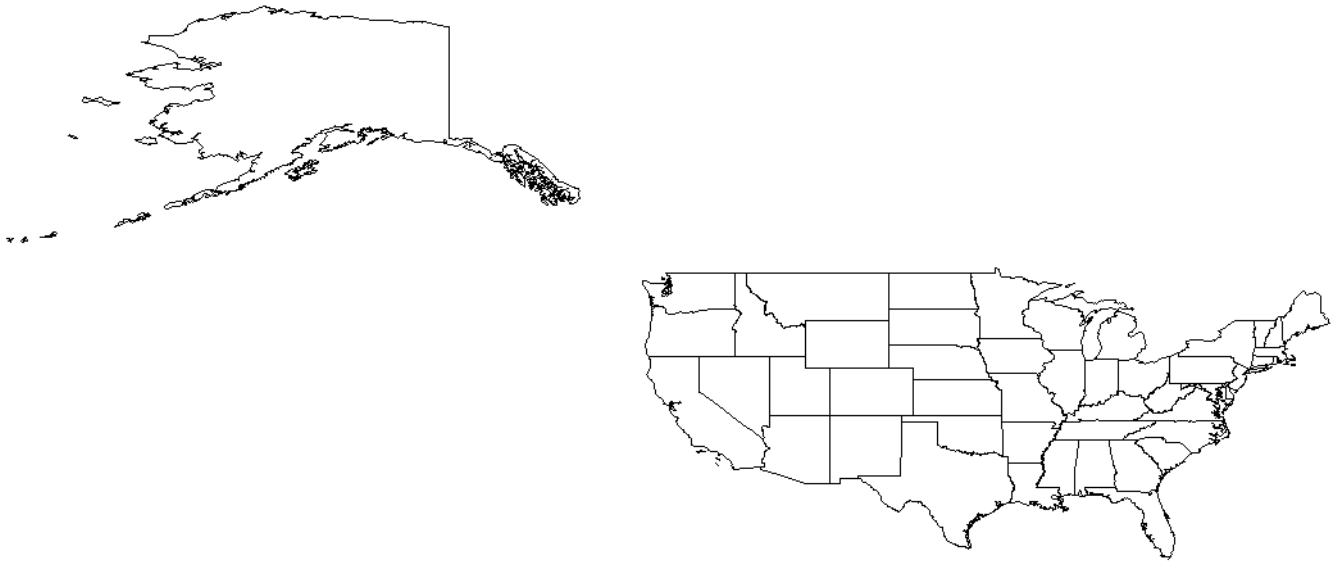
Erfassungsmaßstab	**Anzahl der Stützpunkte**

1:10.000.000	403706
1:50.000.000	60699
1:110.000.000	5143

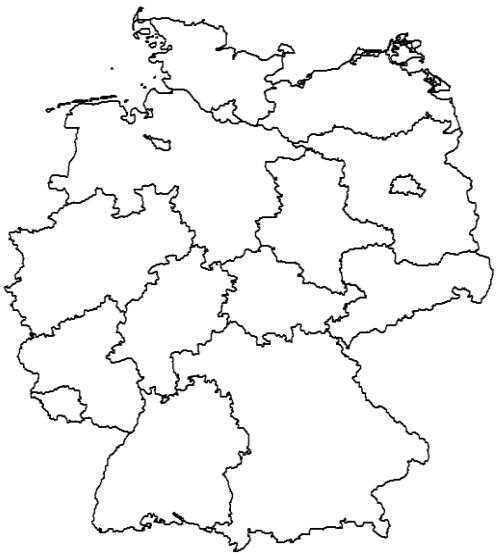
Tabelle 7-1: Unterschiedliche Anzahl der Objekte sowie Punkte für Weltkarten mit verschiedenen Maßstäben

Der Vergleich von Karten setzt voraus, dass sie ein gemeinsames Gebiet zeigen. Ist dies gegeben, so lässt sich der jeweilige Maßstab der Karten aus der Anzahl der Stützpunkte verhältnismäßig beurteilen. Die Anzahl der Stützpunkte für verschiedene Gebiete ist angesichts unterschiedlicher geometrischer Eigenschaften aufgrund verschiedener geografischer Gegebenheiten differierend. Dies illustriert Abbildung 7-3. Es handelt sich um zwei politische Karten. Jeweils eine für die USA und eine für Deutschland. In der Karte der USA werden die Begrenzungen häufig mit geraden Linien dargestellt, wofür wenige Stützpunkte ausreichend sind. Dagegen benötigt die kurvige Begrenzung in der Karte für Deutschland verhältnismäßig mehr Stützpunkte.

Daraus ist ersichtlich, dass beim Vergleich verschiedener Gebiete die Anzahl der Stützpunkte pro Fläche zur Maßstabsinterpretation nicht angebracht ist. Ausnahmen dieser Feststellung können Kartentypen sein, die aufgrund ihres geografischen Inhalts, zu ähnlicher geometrischer Formgebung tendieren. Als Beispiel dazu werden im Folgenden Flusskarten interpretiert.



a) USA



b) Deutschland

Abbildung 7-3: Politische Karten für USA und Deutschland

7.2.1.2 Flusskarten

Der Vergleich von Flusskarten eines nicht-gemeinsamen Gebiets zum Zweck der Erhebung des Detaillierungsgrads ist möglich, weil die Karten eine ähnliche geometrische Formgebung beinhalten. Flussläufe sind generell in ähnlicher Weise gewunden. Sie werden in jeder Generalisierungsstufe mit gekrümmten Linien auf der Karte dargestellt. Aufgrund dieser ähnlichen Linienverläufe kann die Anzahl der Stützpunkte auf Flusskarten verglichen werden, obgleich die Flüsse nicht durch ein gemeinsames Gebiet ziehen.

In der Abbildung 7-4 wird die Anzahl der Stützpunkte pro Kilometer auf jeweils drei Flusskarten in verschiedenen bekannten Erfassungsmaßstäben angezeigt. Die Flusskarten sind nicht für jeden Maßstab vorhanden. Die Anzahl der Stützpunkte für den mittleren Maßstab 1:24.000 ist über 4000. Der Wertebereich für den mittleren Maßstab 1:100.000 liegt im Bereich 1000 – 2000. Für die Maßstäbe 1:1.000.000 sowie 1:2.000.000 sind die Werte im Bereich 200 – 600. Für die kleinen Maßstäbe kleiner als 1:10.000.000 liegt die Anzahl unter 50. Es lässt sich erkennen, dass sich die Wertebereiche für die Anzahl der Stützpunkte abhängig vom jeweils vorliegenden Maßstab verhalten.

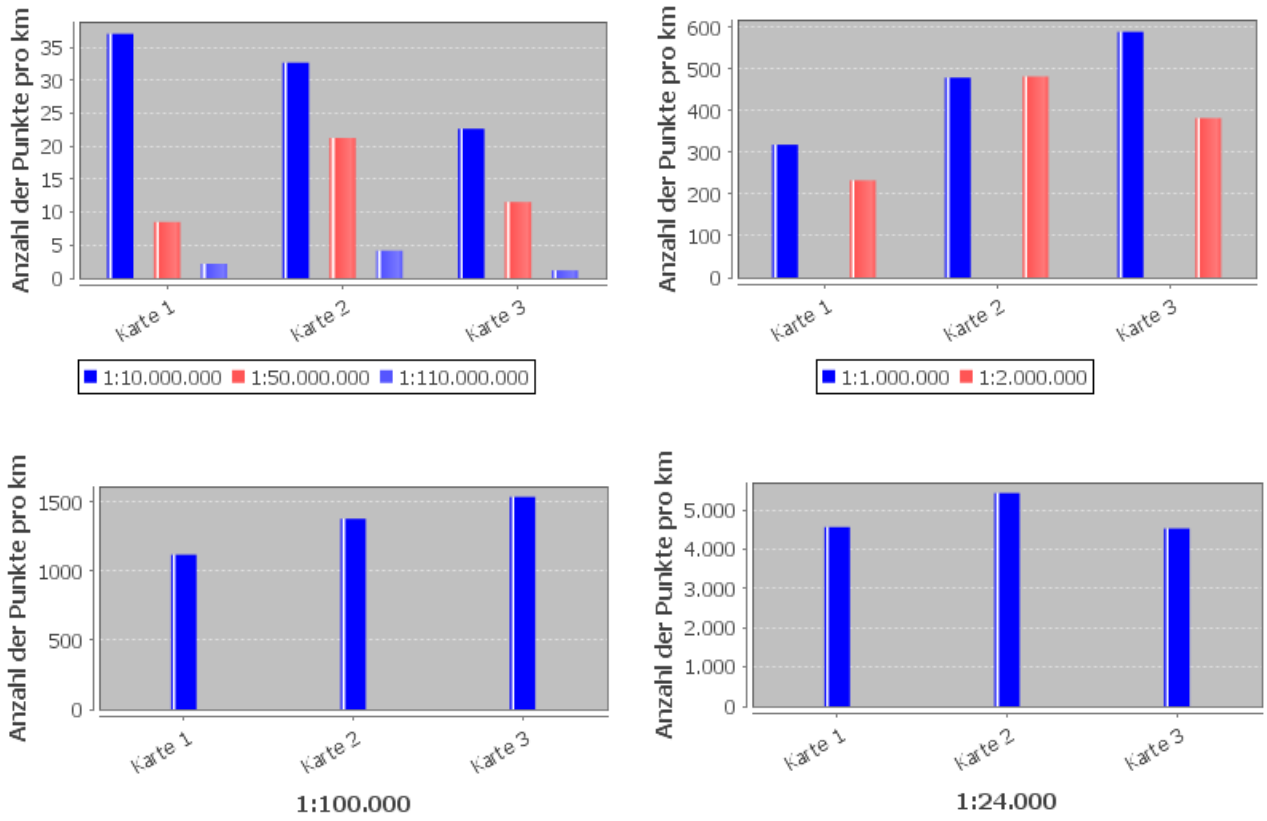


Abbildung 7-4: Anzahl der Stützpunkte pro Kilometer auf Flusskarten

7.2.2 Abstand zwischen Linien

Ein weiterer Aspekt des Detaillierungsgrads ist der Abstand zwischen bestimmten Linien. Der Abstand wird mit zunehmendem Maßstab stetig kleiner. Der Abstand soll die Distanz zwischen parallelen bzw. nahezu parallelen Linien, wie z.B. Höhenlinien oder Gitternetzlinien, die in einer Richtung liegen, sein. Im Folgenden werden die Auswirkungen verschiedener Distanzen zwischen Linien auf den Maßstab betrachtet.

7.2.2.1 Höhenlinienkarten

Die Höhenlinien sind die linearen Verbindungen der Punkte gleicher topografischer Höhe. Der Höhenabstand zwischen den Höhenlinien, auch Äquidistanz genannt, ist maßstabsabhängig. In [Imhof 1965] wird folgende Formel zur Berechnung der Äquidistanz erläutert:

$$A = n \cdot logn \cdot tan\alpha; \quad n = \sqrt{\frac{M}{100} + 1}$$

M: Maßstabszahl

α: Geländeneigung

[Hake et al. 2002] geben in der folgenden Tabelle 7-2 die auf volle Meter abgerundeten Werte der Äquidistanz für die gebräuchlichsten großen und mittleren Maßstäbe.

Maßstab / **Gelände-neigung α_{max}**	**2000**	**5000**	**10.000**	**25.000**	**50.000**	**100.000**	**200.000**
45° (Gebirge)	2	5	10	20	30	50	100
25° (Berg- u. Hügelland)	1	2	5	10	15	25	50
10° (Flachland)	0,5	1	2	2,5	5	10	10

Tabelle 7-2: Ideale Äquidistanzen der Höhenlinien für große und mittlere Kartenmaßstäbe (nach [Hake et al. 2002])

Eine einheitliche Äquidistanz für ein großes Kartenwerk ist schwierig zu realisieren, da die maximale Geländeneigung stark variieren kann. In diesem Fall werden nach Bedarf an flachen Stellen Zwischenhöhenlinien hinzugefügt, wobei die Äquidistanz für flachere Gebietsteile allgemein kleiner gewählt wird [Hake et al. 2002]. Darüber hinaus ist die Äquidistanz im Maßstabsbereich 1:200.000 bis 1:1.000.000 in der Regel nicht einheitlich. Dafür werden nicht-äquidistante Höhenlinien, deren vertikale Abstände mit wachsender Höhe zunehmen, angewendet [Brunner 2003].

Aus der Äquidistanz sowie der Geländeneigung lässt sich der horizontale Abstand zwischen den Höhenlinien, der auf der Karte gemessen werden kann, mit folgender Formel berechnen:

$$H = \frac{A}{tan\alpha}$$

Aus der Berechnung ergeben sich in der Tabelle 7-3 die auf volle Meter abgerundeten Werte der horizontalen Abstände für die gebräuchlichsten großen und mittleren Maßstäbe.

Maßstab / Gelände-neigung α_{max}	2000	5000	10.000	25.000	50.000	100.000	200.000
45° (Gebirge)	16	40	80	160	239	398	796
25° (Berg- u. Hügelland)	14	29	72	144	216	359	719
10° (Flachland)	18	36	72	90	180	360	360

Tabelle 7-3: Ideale horizontale Abstände der Höhenlinien für große und mittlere Kartenmaßstäbe

Aufgrund des Zusammenhangs zwischen den horizontalen Abständen und dem Maßstab kann der Maßstab aus dem Abstand abgeleitet werden. In dieser Arbeit wird der durchschnittliche Abstand zwischen den Höhenlinien auf den Höhenlinienkarten, deren Erfassungsmaßstab bekannt ist, berechnet. Es sind insgesamt 66 Höhenlinienkarten mit bekanntem Erfassungsmaßstab vorhanden. Der durchschnittliche Abstand des entsprechenden Maßstabs wird in der Tabelle 7-4 dargestellt. Der Wert wird abgerundet.

Maßstab	5000	20.000	24.000	50.000	100.000	1.000.000	2.000.000	10.000.000
Durchschnittlicher Abstand (m)	35	145	120	190	500	3600	5300	52000

Tabelle 7-4: Durchschnitt der Abstände zu entsprechenden Maßstäben

Der durchschnittliche Abstand entspricht grob der Zuordnung. Zwischenhöhenlinien sowie nichtäquidistante Höhenlinien auf den Karten verhindern eine noch genauere Zuordnung. Aus den erzielten Ergebnissen lässt sich folgern, dass der Abstand in den Vektorkarten tatsächlich vom Maßstab abhängt. Aus den horizontalen Abständen zwischen den Höhenlinien der Vektorkarten kann der große und mittlere Maßstab nach der Tabelle 7-3 abgeleitet werden. Für die Karten mit kleinem Maßstab ist der Abstand größer als der größte Wert der Tabelle 7-3.

7.2.2.2 Gitternetzkarten

Kartennetze bestehen aus einheitlichen Bezugslinien für die Bestimmung eines Geländepunktes. Die quadratischen Kartennetze dienen dazu, die Koordinaten eines Orts auf der Karte zu entnehmen oder umgekehrt einen Ort auf der Karte aus gegebenen Koordinaten zu finden. Der Abstand der das Gitternetz bildenden Parallelen wird auf den Maßstab der Karte bezogen. Als Beispiel hierfür enthält das

Gauß-Krüger-Netz den Abstand 4 cm auf der Topographische Karte 1:25.000 (TK 25) [Hake et al. 2002], was 1 km in der Natur entspricht.

Wenn der Abstand auf der Gitternetzkarte ermittelt wird, kann der Maßstab daraus erkannt werden. Im Folgenden wird dies beispielhaft mit dem weit verbreiteten UTM-Gitternetz untersucht. In [Esri 2012] werden Abstände zwischen den Gitternetzlinien auf Grundlage der WGS84 UTM-Projektion sowie auf der NAD 1983 UTM-Projektion zu entsprechenden Maßstäben angegeben (Tabelle 7-5).

Maßstab	**500**	**1000**	**24.000**	**50.000**	**250.000**	**500.000**
Abstand	10 m	50 m	1 km	1 km	10 km	10 km

Tabelle 7-5: UTM-Gitterabstand zu Maßstäben

Die Gitternetzkarten mit bekannten Erfassungsmaßstäben werden nach dem Gitterlinienabstand geprüft. Die Karten werden zuerst daraufhin geprüft, ob sie die WGS84 UTM-Projektion oder die NAD 1983 UTM-Projektion besitzen. Eine Karte wird mit Abstand von 1 km zum Maßstab 1: 24.000, drei Karten mit Abstand von 1 km zum Maßstab 1: 50.000 und eine Karte mit Abstand von 1 km zum Maßstab 1: 100.000 erkannt. Die Zusammenhänge des Abstands und Maßstabs entsprechen der Zuordnung in der Tabelle 7-5. Schon mit wenigen Gitternetzkarten kann gezeigt werden, dass die Erkennung des Maßstabs aus dem Abstand der Gitternetzlinien möglich ist.

7.3 Diskussion

Die Detaillierungsgrade für Vektorkarten können von der Genauigkeit der Digitalisierung beeinflusst werden. Dies gilt insbesondere für die Anzahl der Stützpunkte. Der Zeichner kann bei der Erstellung der Vektorkarte entscheiden, wie viele Stützpunkte er für das Zeichnen beispielsweise eines Kreises verwendet. Demzufolge können die Detaillierungsgrade nicht strikt mit den Maßstäben zusammenhängen. Da aber nicht konkrete Maßstäbe, sondern die Maßstabsbereiche ausgewählt werden, ist die Zuordnung der Detaillierungsgrade insgesamt stimmig. Die Untersuchung im vorliegenden Kapitel zeigt, dass Maßstäbe aus Detaillierungsgraden hergeleitet werden können. Für eine weiter reichende Gesetzmäßigkeit des Zusammenhangs müssen größere Mengen an Karten mit verschiedenen Maßstäben untersucht werden.

Neben dem Vergleich von Flusskarten eines nicht-gemeinsamen Gebiets zum Zweck der Erhebung des Detaillierungsgrads ist es möglich, weitere Kartentypen mit ähnlicher geometrischer Formgebung zu berücksichtigen. Beispielsweise kann die Anzahl der Stützpunkte in Küstenlinienkarten verglichen werden, da die Küstenlinienkarten sich aus gekrümmten zerklüfteten Küstenlinien zusammensetzen.

Höhenlinien- und Gitternetzkarten stellen sich häufig als Basiskarten anderer Karten dar. Sie werden mit anderen Karten gleichen Maßstabs wie z.B. politischen Karten oder Natur-Karten etc. in unterschiedliche Layer zusammen dargestellt. Diese zusammengehörigen Shapefiles werden möglicherweise in einer .zip-Datei komprimiert. Der Maßstab der anderen Karten in der .zip-Datei kann somit aus dem Maßstab der Höhenlinienkarten oder Gitternetzkarten abgeleitet werden.

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

In der vorliegenden Arbeit werden automatische Verfahren und Methoden zur Karteninterpretation aufgezeigt. Die Verfahren sollen das implizite Wissen in den Karten automatisch explizit machen, ohne dass dabei menschliche Betrachtung erforderlich ist. Die unterschiedlichen Verfahren in der Arbeit haben die Gemeinsamkeit, dass ihre Realisierung auf Regelmäßigkeiten in den Datensätzen basiert. Aus den Regelmäßigkeiten kann der Maschine beigebracht werden, die gewünschten Informationen aus den Karten aufzudecken.

Die Verfahren konzentrieren sich auf Vektordaten. Es wird ein Webcrawler entwickelt, um die Vektorkarten mit dem Format Shapefile aus dem Internet zu erhalten. Dabei handelt es sich einmal um eine Suche ohne jegliche Einschränkungen. Beim anderen Mal findet die Suche lediglich in einem Server statt. Die zu durchsuchenden Server werden durch Google-Suche vorselektiert. Mit Google wird nach dem Schlüsselwort „shapefile download“ gesucht. Die Suche durch den Webcrawler wird dann auf die resultierenden Webseiten beschränkt, um die Effektivität zu erhöhen. Das Verfahren mit Google-Suche erweist sich bei der Suche von Vektorkarten als effektiver.

Zur Karteninterpretation werden Interpretationsverfahren bezüglich der Kartenobjekte, der Kartentypen sowie des Maßstabs entwickelt. Das erste Verfahren greift die Interpretation der Kartenobjekte auf. Die Objektklassen werden anhand ihrer geometrischen Eigenschaften beschrieben und lassen sich durch diese Charakterisierung voneinander differenzieren. Basierend auf den geometrischen Beschreibungen der Objektklassen wird die Karteninterpretation mittels SOM umgesetzt. Die Objektklassen werden entsprechend ihrer geometrischen Merkmale in Form eines Parametervektors beschrieben. Aus den Parametervektoren werden die Eingabemuster hergeleitet und in der Lernphase der SOM gelernt. Wird danach ein Objekt in die SOM eingegeben, werden seine geometrischen Eigenschaften ermittelt und der entsprechenden Klasse zugeordnet.

Anhand einer Testkarte wird verdeutlicht, dass die Interpretation durch die Auswahl der Parameter beeinflusst wird. Je präziser die Parameter aufgeteilt werden, desto genauer ist die Interpretation. Es wird gezeigt, wie das Ergebnis der Interpretation sich ändert, wenn Parameter geändert werden. Des Weiteren wird diskutiert, wie das Verfahren mit einer anderen Testkarte gleicher Objektklassen funktioniert. Dabei sind die geometrischen Parameter für die zweite Testkarte gültig. Für Karten mit unterschiedlichen Objektklassen muss die Geometrie erneut analysiert werden. Es zeigt sich, dass die Objekte erkannt werden, auch wenn ihre Eingabeparameter nicht exakt mit einem gelernten Muster übereinstimmen. Für den Grad der Übereinstimmung der Eingabeparameter mit den Musterparametern kann die s.g. *Qualität* berechnet werden.

Auch der Kartentyp soll mittels SOM interpretiert werden. Es werden sowohl Kartentypen mit linienförmigen als auch polygonförmigen Objekten untersucht. Auch hier liegt das Hauptaugenmerk auf der Festlegung der jeweiligen Merkmale. Allerdings sind hier nicht wie bei der Interpretation der Kartenobjekte nur die geometrischen Eigenschaften von Bedeutung, sondern es werden auch die topologischen Eigenschaften der Kartentypen berücksichtigt. So ist der Knotentyp ein wichtiges Merkmal für die Kartentypen mit linienförmigen Objekten. Es wird gezeigt, dass unterschiedliche Kartentypen verschiedene Knotentypen besitzen. Sie lassen sich also dadurch voneinander unterscheiden.

Die durch den Webcrawler erworbenen Datensätze werden für die Interpretation der Kartentypen angewandt. Zur Interpretation werden Karten ausgewählt, die durch die bloßen Augen interpretiert werden können. Solche Karten besitzen eigene Merkmale, die von der SOM mit den Mustern verglichen und somit erkannt werden können. Bei der Interpretation des Kartentyps werden außerdem Zusatzinformationen genutzt. Die Zusatzinformationen sind die Dateinamen der Karten sowie der Inhalt der Webseite, auf welcher die Karte gefunden wurde. Zum einen enthält der Dateiname einer Karte treffende Stichwörter, die den Karteninhalt zusammenfassen. Zum anderen enthält die Webseite möglicherweise eine Erläuterung über die Karte. Die Erläuterung kann über ein Text-Mining-Verfahren herausgefunden werden.

Im letzten Teil der Arbeit wird die Interpretation des Kartenmaßstabs untersucht. Dies geschieht einerseits mittels Mehrfachrepräsentation, andererseits mittels der Detaillierungsgrade. Mit der Mehrfachrepräsentation wird ein identisches Objekt in unterschiedlichen realitätsgetreuen Repräsentationen auf der Karte dargestellt. Die Darstellung in Karten größeren Maßstabs bildet die genauere Struktur ab, während die Struktur in Karten kleineren Maßstabs lediglich symbolhaft gezeigt wird. Es wird in der vorliegenden Arbeit anhand eines Beispiels, dem Kreisverkehr, gezeigt, wie sich der Maßstab aus der entsprechenden Repräsentation herleiten lässt. Des Weiteren wird der Zusammenhang der verschiedenen Maßstäbe zu den Detaillierungsgraden der Darstellungen untersucht. Auf der Karte größeren Maßstabs werden die Objekte detaillierter, auf der Karte kleineren Maßstabs weniger detailliert dargestellt. Die Erscheinungsform ist dabei identisch. Die Ableitung der Maßstäbe wird hinsichtlich der Anzahl der Stützpunkte sowie des Abstands von Linien diskutiert.

8.2 Ausblick

Im Bereich der Karteninterpretation, insbesondere der Interpretation der Kartenobjekte, der Kartentypen sowie des Kartenmaßstabs kann die vorliegende Arbeit einen Beitrag leisten. In zukünftigen Arbeiten kann untersucht werden, eine semantische Suche basierend auf der Interpretation der Karten zu ermöglichen. Hierfür können zahlreiche Vektorkarten aus dem Internet heruntergeladen, in einem Repository gespeichert und online zur Verfügung gestellt werden. Damit der Nutzer direkten Zugang zu den durch die Interpretation explizit gewordenen Karteninformationen hat, kann eine semantische Suche nach den Karten Abhilfe schaffen.

Hierzu soll versucht werden, die Interpretation weiterer Kartentypen sowie des Maßstabs zu erweitern. In der vorliegenden Arbeit wurden lediglich bestimmte Teile der Karteninterpretation untersucht. Ferner können andere Aspekte der Interpretation erforscht werden. Als ein Beispiel können die Gebiete, auf die die Karten abbilden, bestimmt werden. Um dies zu ermöglichen, sollen zunächst die Koordinaten des Begrenzungsrechtecks einer Vektorkarte ermittelt werden. Die Ermittlung der Koordinaten auf Vektorkarten ist einfach. Außerdem soll eine geographische Datenbank zur Verfügung stehen. In der Datenbank werden die Koordinaten des Begrenzungsrechtecks von zahlreichen Städten, Staaten, Ländern etc. gespeichert. Es existieren mittlerweile auch freie geographische Datenbanken wie z.B. GeoNames und OpenGeoDB. Die ermittelten Koordinaten der Karte können mit den Koordinaten in der Datenbank verglichen werden und die Karte kann somit dem entsprechenden Gebiet zugewiesen werden.

Nachdem die Karteninterpretation erweitert wurde, kann eine semantische Suche der Karten entworfen werden. Dafür können die Karten mit einer Auszeichnungssprache aus den Methoden des semantischen Webs wie z.B. RDF (Resource Description Framework) für Ontologien beschrieben werden. Ein RDF Modell besteht aus Ressourcen, Attributen der Ressourcen und Werten der Attribute. Der Zusammenhang von Ressourcen, Attributen und Werten bilden ein Ressource-Attribut-Wert-Tripel. Das Tripel wird auch RDF-Statement genannt. Die drei Teile Ressource, Attribut und Wert werden auch als Subjekt, Prädikat und Objekt bezeichnet. Eine Karte ist hier eine Ressource. Sie kann mittels einem URI (Universal Resource Identifier) identifiziert werden. Für eine Karte im Repository kann das URI beispielsweise mit dem Link präsentiert werden. Ein Attribut kann eine Ressource in einer speziellen Perspektive beschreiben. Die Attribute können mit dem RDF-Schema (RDF Vocabulary Description Language) definiert werden. Für eine Karte sind Attribute beispielsweise der Kartentyp, der Maßstab oder das Gebiet mit den Beispielwerten Flusskarte, kleiner Maßstab und Deutschlandkarte.

Um eine Ressource abzufragen, kann eine Abfragesprache aus den Technologien des semantischen Webs wie beispielsweise SPARQL (Protocol And RDF Query Language) eingesetzt werden. SPARQL ist eine an SQL angelehnte Abfragesprache zur Traversierung von RDF-Tripeln. SPARQL Variablen sind mit RDF-Elementen gebunden und bilden die Abfragen in Form von einem SELECT-Statement wie in der Datenbanksprache SQL (Structured Query Language). In dem Statement kann man Suchbedingungen zusammentragen, um gewünschte Karten zu bekommen.

Damit der Nutzer gewünschte Karten in natürlicher Sprache suchen kann, soll die natürliche Sprache in die Abfragesprache SPARQL translatiert werden. Der in natürlicher Sprache eingegebene Suchtext wird von einem Analyseprogramm gefiltert und extrahiert. Der erhaltene Inhalt bezieht sich möglicherweise auf die in der Ontologie modellierten Elemente. Ein Agent kann die extrahierten Elemente mit den RDF-Tripeln vergleichen und daraus die Logik ableiten. Mit der Logik werden die SPARQL-Abfragen erstellt. Als Beispiel wird der folgende Suchtext eingegeben: „finde alle großmaßstäbigen

Straßenkarten in Baden-Württemberg“. Der Suchtext wird nach der Semantik analysiert. Die Elemente „Kartentyp“ mit dem Wert „Straßenkarte“, „Maßstab“ mit dem Wert „groß“ und „Gebiet“ mit dem Wert „Baden-Württemberg“ können abgeleitet werden. Diese Elemente und Werte werden in der SPARQL-Abfrage zusammengetragen und somit werden entsprechende Karten als Ergebnis geliefert.

Anhang A Breitensuche und Tiefensuche

Anhang A.1 Breitensuche

Breitensuche (breadth-first search, BFS) ist ein Verfahren zur Suche von Knoten in einem Graphen. Ein Startknoten *v* wird ausgewählt. Alle direkt nachfolgenden Knoten werden dann einer Warteschlange hinzugefügt, falls dies noch nicht geschehen ist. Nachdem alle nachfolgenden Knoten des Startknotens bearbeitet wurden, wird der erste Knoten aus der Warteschlange entnommen. Das Verfahren wiederholt sich solange, bis alle Knoten in diesem Graphen betrachtet werden.

```
1. BFS(v):
2.      queue.push(v)
3.      Knoten v als gefunden markieren
4.      while queue ist nicht leer
5.          v = queue.pop()
6.          for each Nachfolge-Knoten x von v do
7.               if x ist als nicht gefunden markiert
8.               queue.push(x)
9.               x als gefunden markieren
```

Anhang A.2 Tiefensuche

Im Vergleich zur Breitensuche bearbeitet die Tiefensuche (depth-first search, DFS) nicht zuerst alle direkt nachfolgenden Knoten, sondern folgt einem Pfad in die Tiefe. Zunächst wird ein Startknoten *v* ausgewählt und der erste Pfad dieses Knotens wird betrachtet. Danach wird der erste Pfad des gegenüberliegenden Knotens betrachtet, falls dieser noch nicht bearbeitet wurde. Der Prozess wiederholt sich solange, bis der gesuchte Knoten gefunden wird oder der ganze Graph durchsucht wurde. Die Tiefensuche lässt sich in rekusiver und nicht rekusiver Implementierung unterscheiden:

- Eine rekursive Implementierung (Vergleich [Goodrich & Tamassia 2001])

```
1  DFS(v):
2      Knoten v als gefunden markieren
3      for each benachbarte Pfade von v nach w do
4          if Knoten w ist nicht als gefunden markiert then
5              DFS(G,w) rekursiv aufrufen
```

- Eine nicht rekursive Implementierung (Vergleich [Kleinberg & Tardos 2006])

```
1  DFS' (v):
2      S ist ein stack
3      S.push(v)
4      while S ist nicht leer
5            v ← S.pop()
6            if v ist nicht als gefunden markiert:
7                v als gefunden markieren
8                for each benachbarte Pfade von v nach w do
9                    S.push(w)
```

Anhang B. Aktivierungsfunktionen

Anhang B.1 Schwellenwertfunktion

Bei der Schwellenwertfunktion gibt es nur zwei Zustände der Aktivierungsstufe: 0 oder 1. Die resultierende Summe der Multiplikation der Eingänge und Gewichtungen wird in die Schwellenwertfunktion $f(net)$ eingesetzt [Rojas 1993]:

$$f(net) = \begin{cases} 1: & falls\ net \geq S \\ 0: & sonst. \end{cases}$$

$$net = \sum_{i=1}^{n} x_i w_i : Netto - Eingang, die\ Summe\ der\ gewichteten\ Eingangssignale$$

$\vec{x}$: *Eingabevektor*

$\vec{w}$: *Gewichtung*

Je nachdem, ob die Summe den Schwellenwert S überschreitet, nimmt der Ausgang die Werte null oder eins an und gibt dies weiter. Der Funktionsverlauf ist wie folgt graphisch dargestellt.

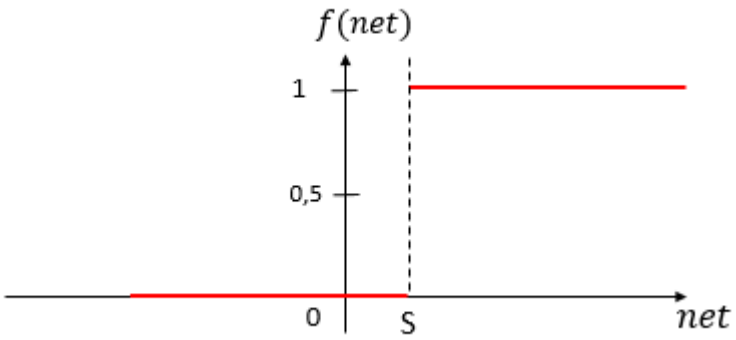


Abbildung B.1: Schwellenwertfunktion

Anhang B.2 Sigmoide Funktion

Eine sigmoide Funktion ist eine mathematische Funktion mit einem S-förmigen Graphen (siehe Abbildung B.2). Die sigmoide Funktion ist differenzierbar. Ein kontinuierliches Ausgangssignal kann geliefert werden.

$$f(net) = \frac{1}{1 + e^{net}}$$

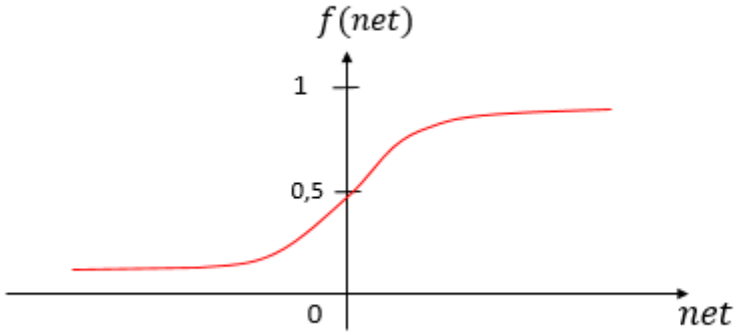


Abbildung B.2: Sigmoide Funktion

Anhang B.3 Lineare Funktion

Bei der linearen Funktion ist der Zusammenhang zwischen dem Eingang und der Aktivierungsstufe linear. Die lineare Funktion stellt die einfachste Aktivierungsfunktion dar. Der Funktionsverlauf ist in Abbildung B.3 graphisch dargestellt.

$$f(net) = net$$

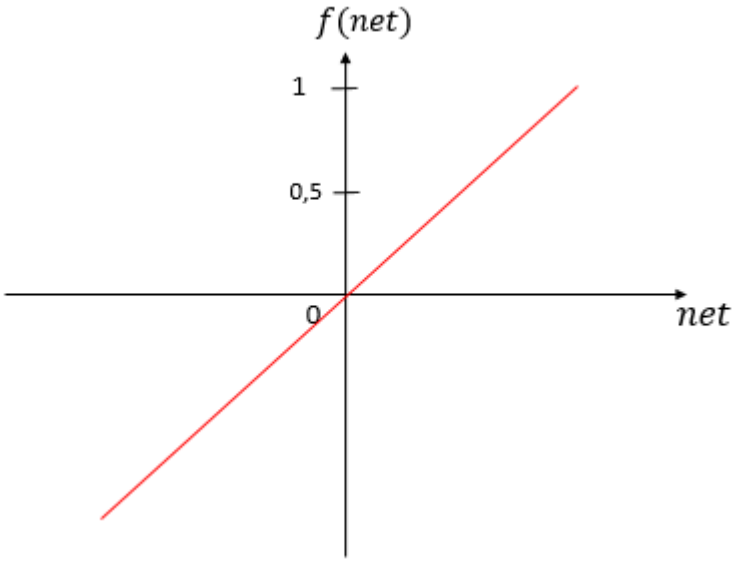


Abbildung B.3: Lineare Funktion

Anhang C. Lernregeln

Anhang C.1 Hebb-Lernregel

Donald Hebb stellte 1949 die Hebb-Lernregel auf, welche sich als Grundlage für viele weitere Lernregeln auszeichnet. Wenn ein Neuron a_i und ein anderes Neuron a_j gleichzeitig und wiederholend aktiv sind, dann wird das Gewicht w_{ij} zwischen den beiden Neuronen verändert. Die Hebb-Lernregel wird als Formel wie folgt zusammengefasst:

$$\Delta w_{ij} = \varepsilon a_i a_j$$

$$\varepsilon: Lernrate$$

Anhang C.2 Delta-Lernregel

Bei der Delta-Lernregel wird das Gewicht proportional zur Differenz der tatsächlich beobachten Ausgabe a_i eines Neurons zu seiner gewünschten Ausgabe a_i' geändert. Die Differenz kann wie folgt dargestellt werden:

$$\delta = a_i' - a_i$$

Das Gewicht wird mit einem zufälligen Wert initialisiert. Die Differenz der Ausgabe wird in jedem Lernschritt für die Eingabe *x* gebildet. Die Gewichtsänderung ergibt sich aus der Multiplikation der Eingabe, der Lernrate und dieser Differenz:

$$w_{neu} = w_{alt} + \Delta w$$

$$\text{mit } \Delta w = \varepsilon * \delta * x$$

$$\varepsilon: Lernrate$$

In dieser Formel sind drei Möglichkeiten enthalten:

- Die beobachtete Ausgabe ist zu niedrig: Die Aktivität wird vergrößert, indem die Gewichte für eine positive Eingabe erhöht und für eine negative Eingabe gesenkt werden.
- Die beobachtete Ausgabe ist zu groß: Die Aktivität wird verkleinert, indem die Gewichte bei den Verbindungen mit positiver Eingabe geschwächt und mit negativer Eingabe gestärkt werden.
- Die beobachtete und gewünschte Ausgabe sind identisch: Die Gewichte werden nicht verändert.

Anhang C.3 Backpropagation

Die oben genannten Lernregeln funktionieren nur bei neuronalen Netzen ohne versteckte Neuronen, welche sich sonst zwischen den Eingabeneuronen und Ausgabeneuronen befinden. Die Backpropagation erlaubt das Trainieren von mehrschichtigen neuronalen Netzen. Die gewünschte Ausgabe ist nur für die Ausgabeschicht, nicht aber für die versteckte Schicht bekannt. Somit kann der Fehler für Neuronen der versteckten Schicht nicht ermittelt werden. Aus diesem Grund berechnet das neuronale Netz die Gewichtsveränderung in zwei Schritten:

- Forward: Der Fehler wird durch das Vergleichen des tatsächlich ermittelten Wertes mit dem gewünschten Wert bestimmt. Als Fehlerfunktion wird der quadratische Fehler verwendet:

$$E = \sum_k (t_k - o_k)^2$$

$$t_k: gewünschter\ Ausgang\ des\ Neurons\ k$$

$$o_k: tatsächlicher\ Ausgang\ des\ Neurons\ k$$

- Backward: Der Fehler geht von der Ergebnisschicht aus und breitet sich Schicht für Schicht durch das neuronale Netz in rückwärtiger Richtung bis zur Eingabeschicht aus. Die Gewichte werden dabei modifiziert.

Im Folgenden soll die Modifikation der Gewichte nach [Pao 1989] hergeleitet werden. Die folgende Abbildung C.1 stellt ein solches Netzwerk dar.

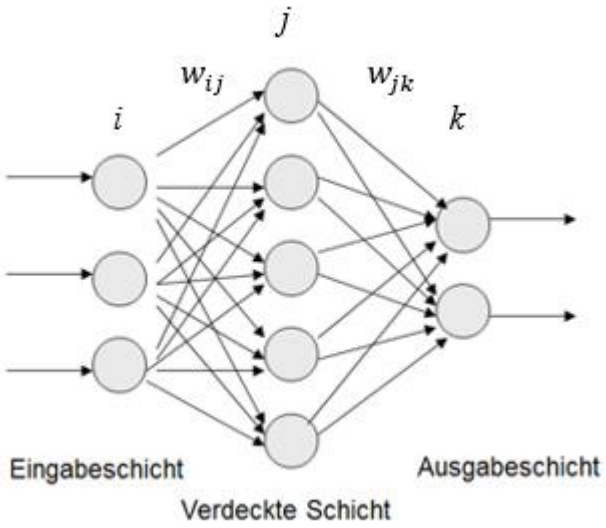


Abbildung C.1: Beispiel eines mehrschichtigen neuronalen Netzes

Die Gewichte werden so verändert, dass der Fehler möglichst klein ausfällt, also absteigend in Richtung des Gradienten des Fehlers:

$$w_{ijneu} = w_{ijalt} + \Delta w$$

$$\Delta w_{ij} = \varepsilon \frac{\partial E}{\partial w_{ij}}$$

$$\varepsilon: Lernrate$$

Mit Hilfe der Kettenregel kann die partielle Ableitung nach dem Netto-Eingang *net* und Neuronausgang *o* eingeführt werden. Hierbei wird unterschieden zwischen:

- Neuron *k* in der Ausgangsschicht

$$\Delta w_{jk} = \varepsilon \frac{\partial E}{\partial w_{jk}} = \varepsilon \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial w_{jk}}$$

Der Fehler ergibt sich als abgeleiteter mittlerer quadratischer Fehler. Zur Erleichterung der Rechnung wird der Faktor $\frac{1}{2}$ eingeführt:

$$E = \frac{1}{2}(t_k - o_k)^2 \qquad \frac{\partial E}{\partial o_k} = t_k - o_k$$

Mit der Aktivierungsfunktion $f(net)$ gilt für die verbleibenden Ableitungen:

$$o_k = f(net_k) \qquad \frac{\partial o_k}{\partial net_k} = f'(net_k)$$

$$net_k = \sum_j w_{jk} o_j \qquad \frac{\partial net_k}{\partial w_{jk}} = o_j$$

Die Gewichtsänderung lässt sich wie folgt zusammenfassen:

$$\Delta w_{jk} = \varepsilon * (t_k - o_k) * \ f'(net_k) * o_j$$

- Neuron *j* in internen Schichten:

$$\Delta w_{ij} = \varepsilon \frac{\partial E}{\partial w_{ij}} = \varepsilon \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}}$$

Da der Fehler nicht direkt abgestimmt werden kann, wird für den Faktor $\frac{\partial E}{\partial o_j}$ auf den Fehler der nachfolgenden Schicht zurückgegriffen:

$$\frac{\partial E}{\partial o_j} = \delta_j = \sum_k \delta_k w_{jk}$$

$$\delta_k = (t_k - o_k) * f'(net_k)$$

Die Gewichtsänderung lässt sich wie folgt zusammenfassen:

$$\Delta w_{ij} = \varepsilon \sum_k \delta_k w_{jk} * \ f'(net_j) * o_i$$

Anhang D. Netztypen

Anhang D.1 Perzeptron

Der Perzeptron Lernalgorithmus wurde 1958 von F. Rosenblatt entwickelt. Das ursprüngliche Perzeptron besteht aus einem einzelnen Neuron. Das Prinzip wird heute zu verschiedenen Kombinationen erweitert. Dabei handelt sich um einschichtige und mehrschichtige Perzeptrons. Die Neuronen innerhalb einer Schicht sind nicht untereinander verbunden, sondern meist mit den Neuronen der folgenden Schicht verbunden. Die Abbildung D.1 zeigt ein einfaches Perzeptron mit einem Eingabe-, zwei verdeckten und drei Ausgabeneuronen.

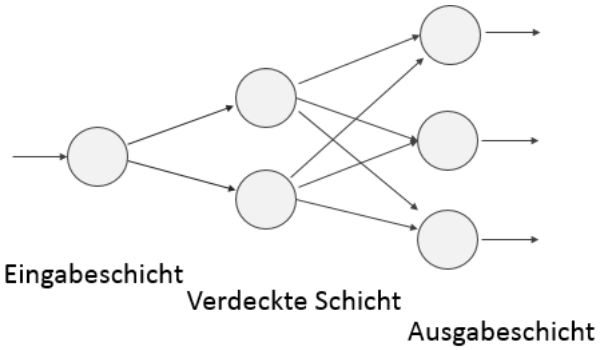


Abbildung D.1: Beispiel Perzeptron

Anhang D.2 Pattern Associator

Das Pattern Associator Netz besitzt keine versteckten Neuronen. Das Netz besteht lediglich aus Ein- und Ausgabeschicht. Als Lernmethoden wird die Hebb-Lernregel oder die Delta-Lernregel eingesetzt. Das Netz kann Muster erkennen, indem es die Assoziationen zwischen verschiedenen Reizpaaren bildet. Die Abbildung D.2 zeigt ein Beispiel des Pattern Associator Netzes mit zwei Eingabe- und drei Ausgabeneuronen.

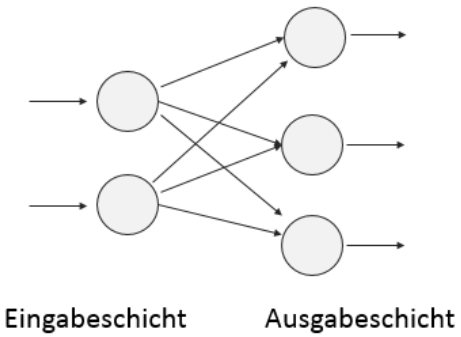


Abbildung D.2: Beispiel eines Pattern Associator Netzes

Anhang D.3 Hopfield-Netz

Das Hopfield-Netz wurde 1982 von J. Hopfield entwickelt. Das Netz unterscheidet sich von anderen neuronalen Netzen, da es keine Eingabe- und Ausgabeschicht enthält. Alle Neuronen sind sowohl Eingabe- als auch Ausgabeneuronen, sie sind mit jedem Neuron, außer sich selbst, verbunden. In Hopfield-Netzen sind die Gewichte symmetrisch, d.h. es gilt $w_{ij} = w_{ji}$ (Abbildung D.3).

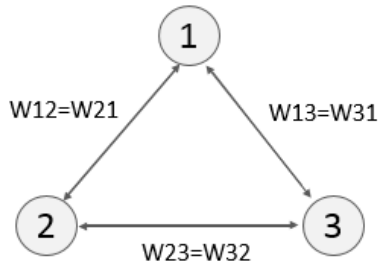


Abbildung D.3: Gewichte in einem Hopfield-Netz

Das Hopfield-Netz benutzt die Hebb-Lernregel als Lernstrategie. Bei der Arbeitsweise von Hopfield-Netzen lassen sich unterscheiden, ob die Gewichte synchron oder asynchron geändert werden sollen.

- Synchron: Der Zustand aller Neuronen wird gleichzeitig aktualisiert
- Asynchron: Nur ein einziges Neuron wird ausgewählt und sein Zustand geändert.

Anhang E. Eigenschaften neuronaler Netze

Anhang E.1 Vorteile neuronaler Netze

- Lernfähigkeit: Neuronale Netze werden mit einer großen Klasse von Trainingsmustern durch spezielle Lernverfahren trainiert. Sie können die Ausgaben schneller als fest programmierte Algorithmen geänderten Eingaben anpassen.

- Parallelität: Neuronale Netze können die notwendigen Berechnungen der Eingabe gleichzeitig durchführen. Sie sind daher für eine Simulation auf parallelen Rechnern sehr geeignet.

- Verteilte Wissensrepräsentation: Die Speicherung von Informationen in neuronalen Netzen ist verteilt. Das Wissen ist in vielen Gewichten gespeichert, so dass eine parallele Verarbeitung möglich ist.

- Höhere Fehlertoleranz: Durch die verteilte Wissensrepräsentation besitzen neuronale Netze eine höhere Fehlertoleranz als herkömmliche Algorithmen beim Absterben einzelner Neuronen oder Verbindungen zwischen Neuronen.

- Teilweise biologische Plausibiltät: Neuronale Netze besitzen aufgrund der parallelen Verarbeitung und verteilten Speicherung eine Ähnlichkeit zum menschlichen Gehirn. Das menschliche Verhalten kann daher gut simuliert werden.

- Robustheit gegenüber Störungen: Richtig trainierte neuronale Netze reagieren auf Störungen oder verrauschte Daten weniger empfindlich als herkömmliche Algorithmen.

- Aktive Repräsentation: Neuronale Netze bringen eine aktive Repräsentation hervor. Dabei werden die Repräsentationen der Information aus den Gewichten miteinbezogen.

Anhang E.2 Nachteile neuronaler Netze

- Wissenserwerb ausschließlich durch Lernen: Die verteilte Information der neuronalen Netze macht es schwer, dem Netz ein basiertes Grundwissen mitzugeben. Die meinsten neuronalen Netze können lediglich durch das Lernen ihre Gewichte bestimmen.

- Keine Analyse möglich: Neuronale Netze können nicht wie ein Expertensystem ihr Wissen bzw. die Information analysieren.

- Großer Rechenaufwand: Neuronale Netze erfordern einen höheren Rechenaufwand zur Lösung als herkömmliche Algorithmen. Viele Lernverfahren neuronaler Netze sind sehr zeitaufwendig. Dies trifft vor allem auf Netze zu, deren Ebenen vollständig miteinander verknüpft sind.

Anhang F. Qualität der SOM

Die folgende Abbildung zeigt die unterschiedlichen Darstellungen der SOM mit verschiedenen Größen. Ab der Größe 10x10 werden die Klassen auf der SOM gleichmäßig verteilt.

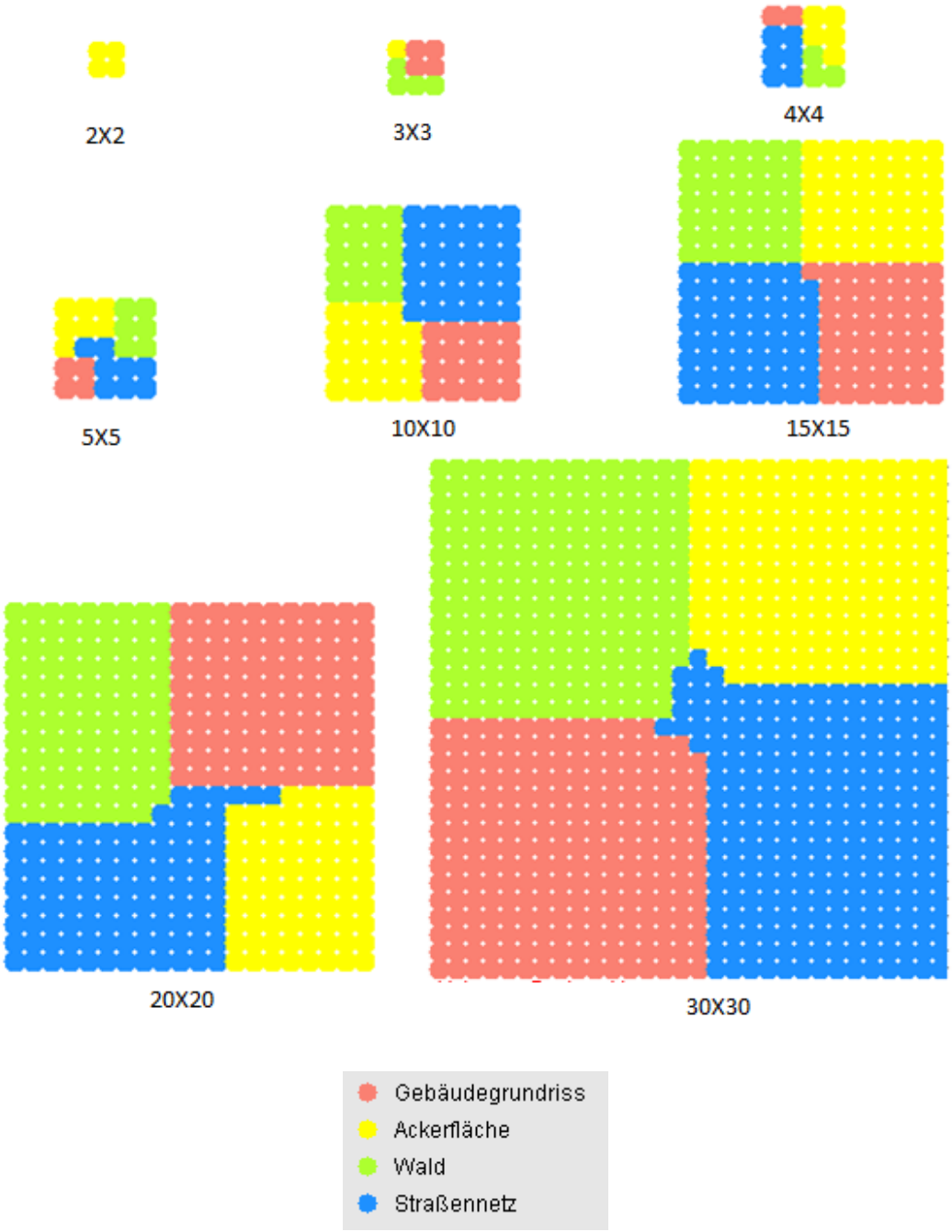


Abbildung F.1: Darstellungen der SOM mit verschiedenen Größen

Anhang G. Ergebnissbeispiele der Interpretation des Kartentyps

Anhang G.1 Karten mit linienförmigen Objekten

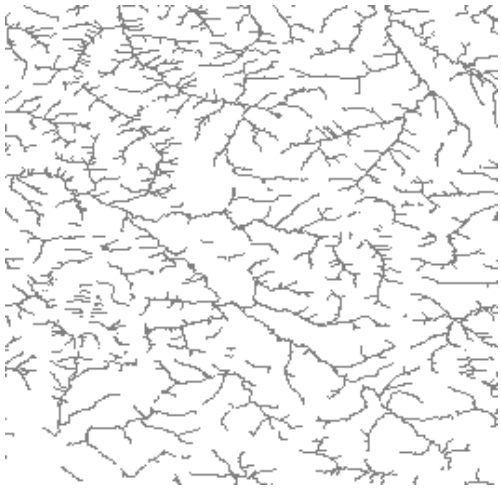
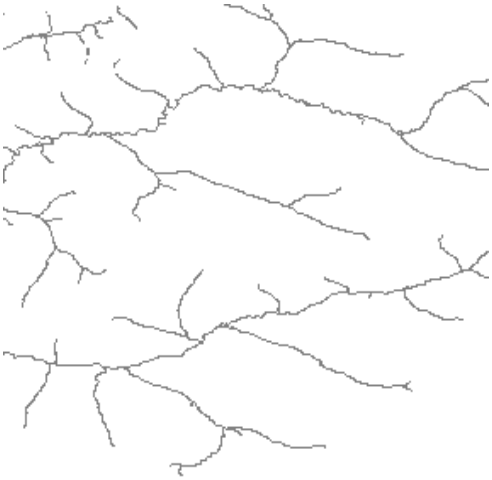
Straßenkarte

- Richtig erkannte Straßenkarten

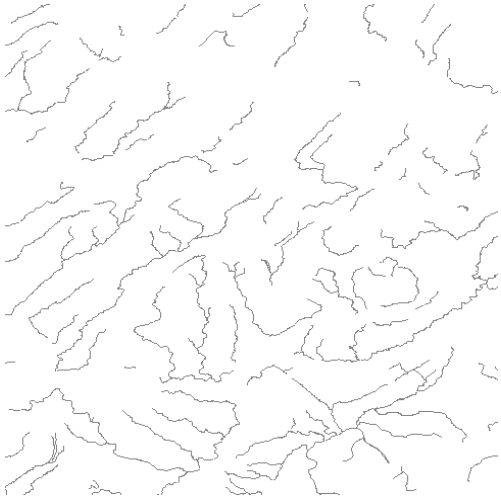
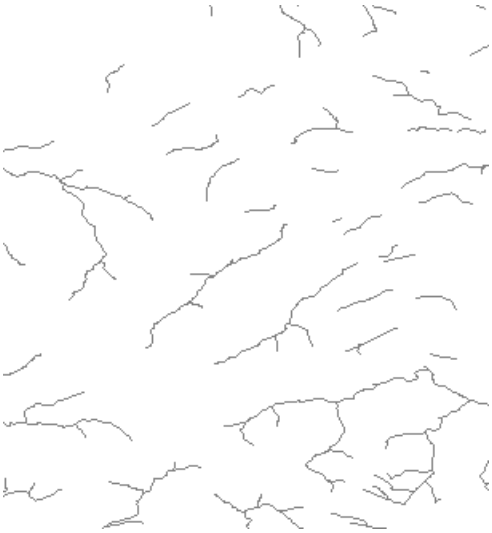

- Aufgrund mangelnder rechter Winkel falsch erkannte Straßenkarten


Flusskarte

- Richtig erkannte Flusskarten

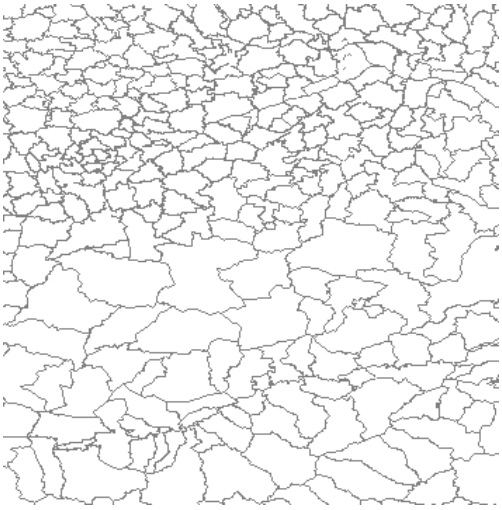



- Aufgrund mangelnder FRK-Knoten (Grad 2) falsch erkannte Flusskarten

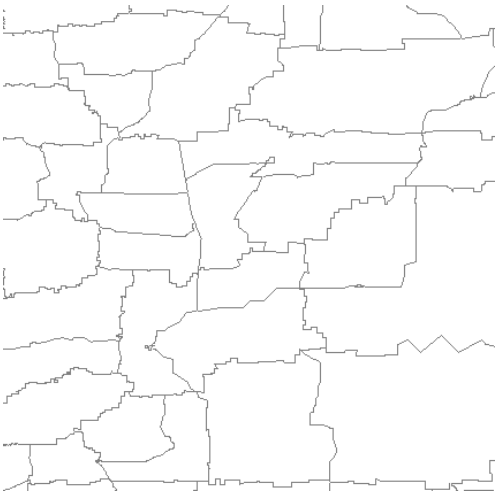



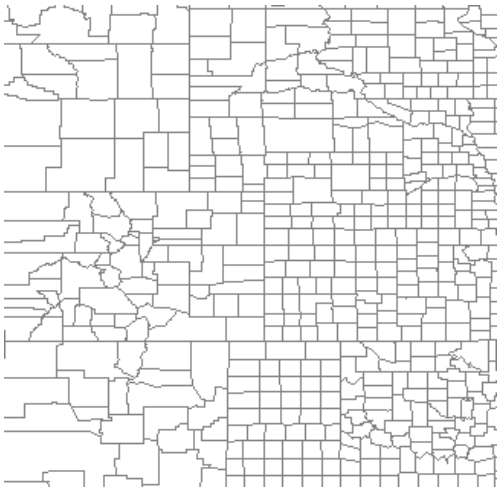
Administrativ-Karte

- Richtig erkannte Administrativ-Karten



- Aufgrund vieler rechter Winkel falsch erkannte Administrativ-Karten



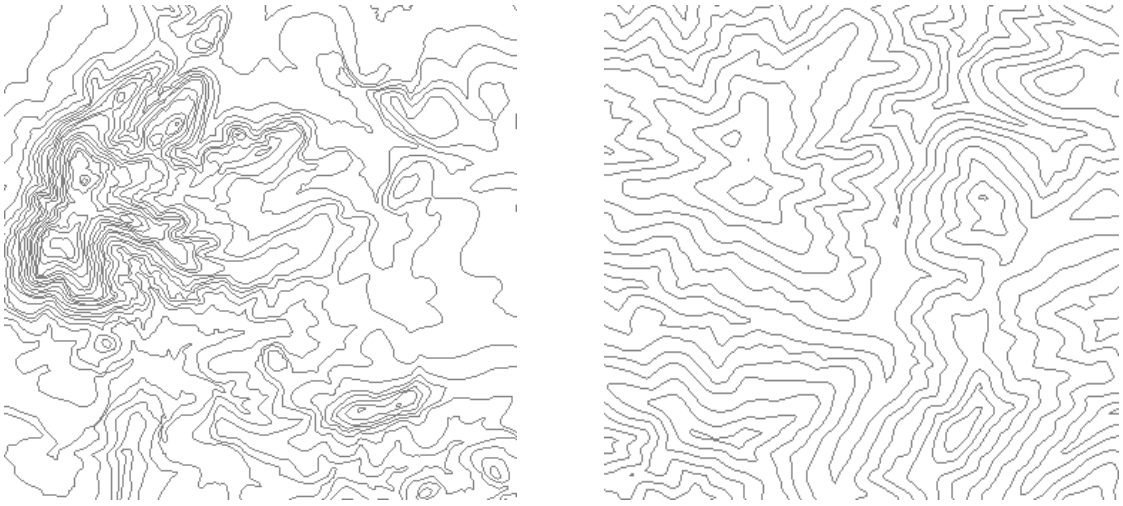


Küstenlinienkarte

- Küstenlinienkarten werden richtig erkannt

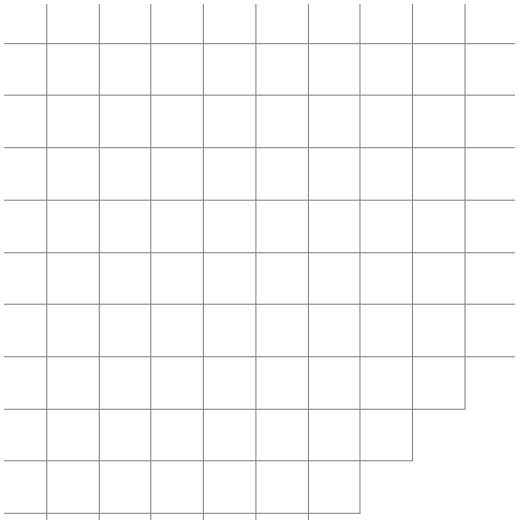
Höhenlinienkarte

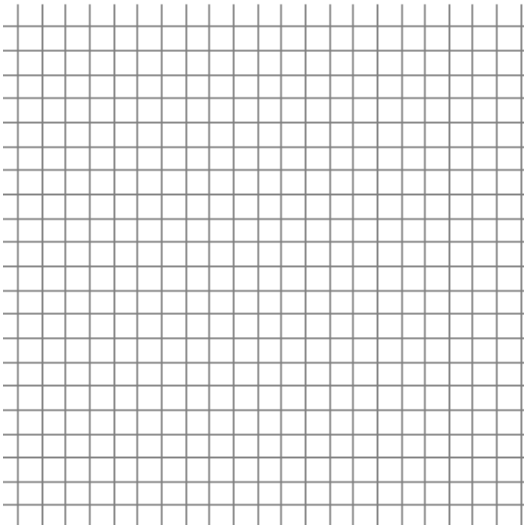
- Höhenlinienkarten werden richtig erkannt



Gitternetzkarte

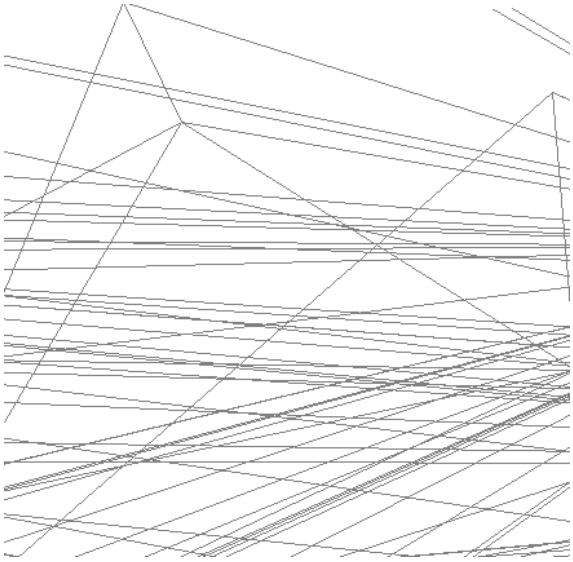
- Gitternetzkarten werden richtig erkannt





Fluglinienkarte

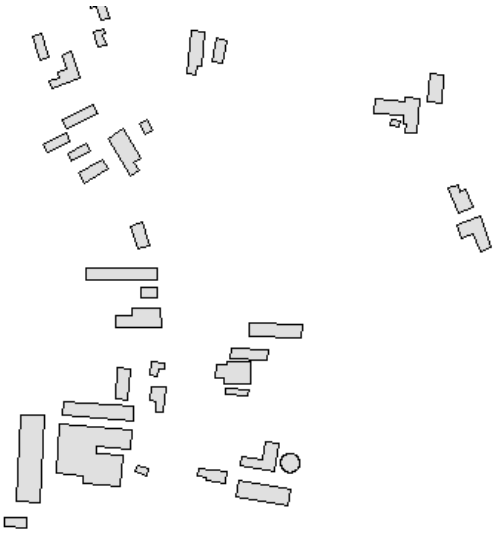
- Fluglinienkarten werden richtig erkannt

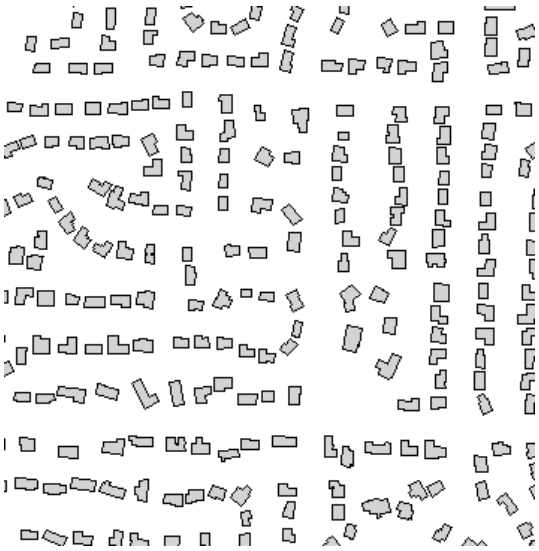


Anhang G.2 Karten mit polygonförmigen Objekten

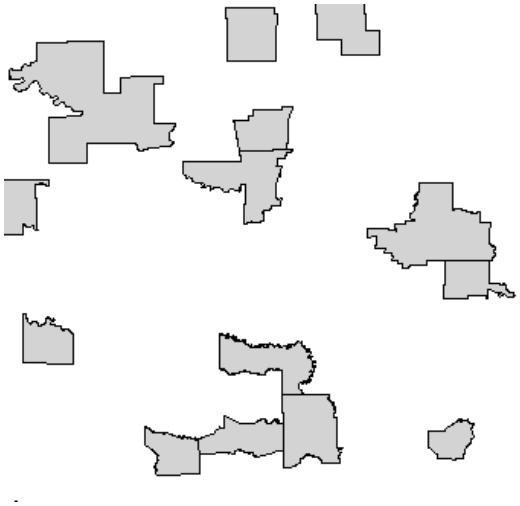
Gebäudegrundrisskarte

- Richtig erkannte Gebäudergrundrisskarten





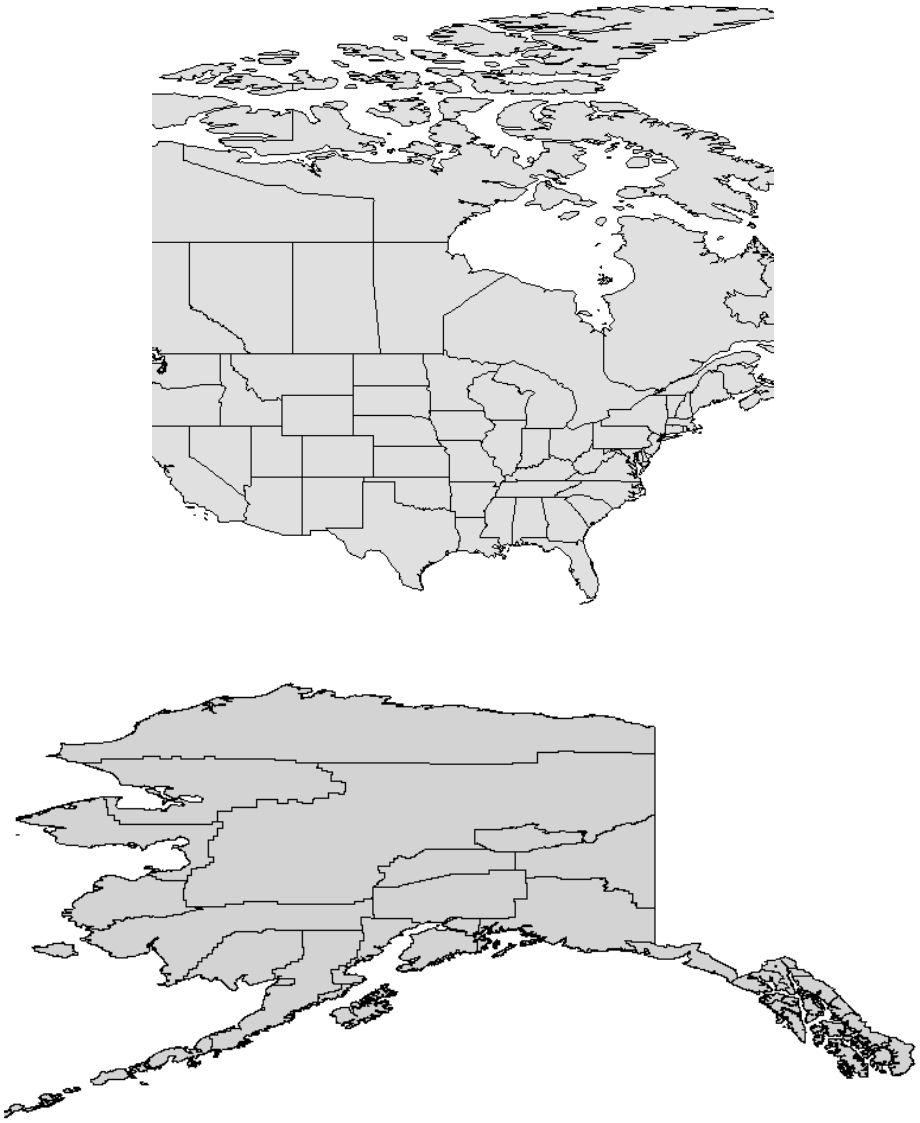
- Aufgrund mangelnder rechter Winkel falsch erkannte Gebäudegrundrisskarten



Administrativ-Karte

- Richtig erkannte Administrativ-Karte

- Aufgrund mangelnder Nachbarschaft falsch erkannte Administrativ-Karten

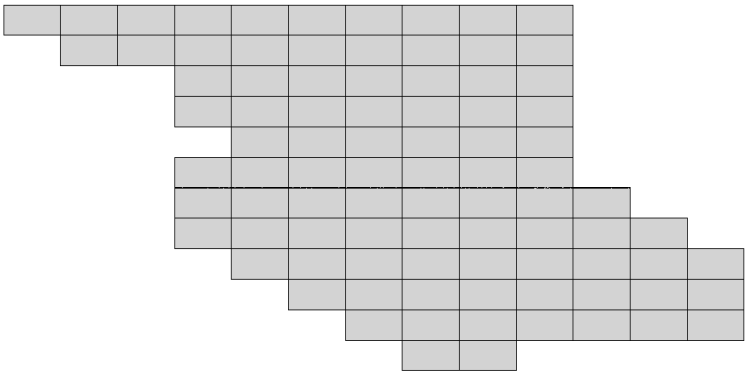


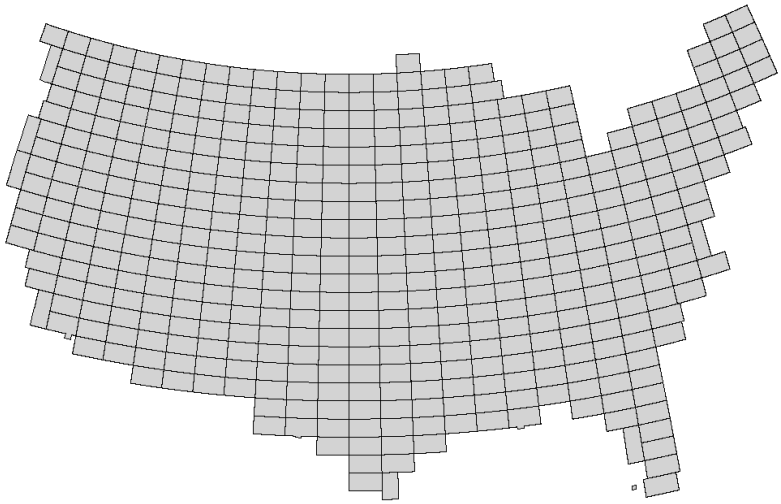
Stadtplan

- Stadtpläne werden richtig erkannt

Gitternetzkarte

- Gitternetzkarten werden richtig erkannt



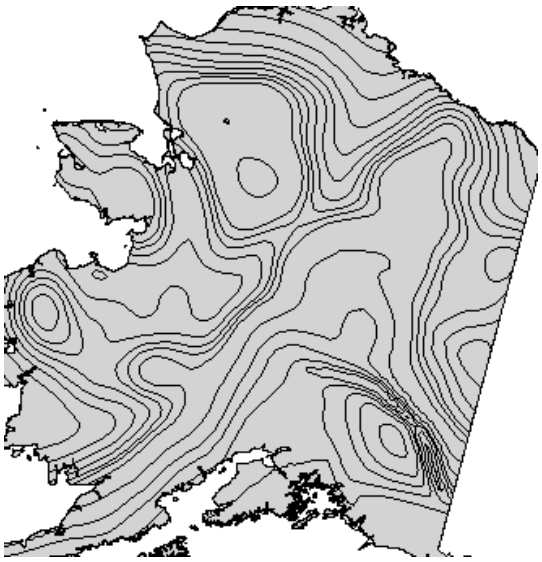


Natur-Karte

- Natur-Karten werden richtig erkannt

Höhenlinienkarte

- Höhenlinienkarten werden richtig erkannt



Anhang H. Beispiele von Karten mit Kreisverkehr

Abbildung E.1: Beispiele von Karten mit Kreisverkehr (Kreisverkehr rot markiert)

Literaturverzeichnis

Allgemeiner Deutscher Automobilclub e.V., Ressort Verkehr (Hg.) [2010]: *Der Kreisverkehr. Informationen, Regeln, Tipps*. http://www.adac.de/_mmm/pdf/rv_kreisverkehr_flyer_0810_27621.pdf.

Agrawal, R., J. Gehrke, D. Gunopulos & P. Raghavan [2005]: *Automatic Subspace Clustering of High Dimensional Data*. In: Data Mining and knowledge discovery, Vol. 11, pp. 5-33.

Anders, K.-H. [2003]: *A hierarchical graph-clustering approach to find groups of objects*. In: ICA Commission on Map Generalization, technical Paper at the Fifth Workshop on Progress in Automated Map Generalization, IGN, Paris (2003).

Anders, F. [2007]: *Mustererkennung in Straßennetzwerken – Verfahren zur Interpretation von Vektordaten*. Dissertation, Leibniz Universität Hannover, Deutsche Geodätische Kommission, Reihe C, Heft Nr. 607, 2007.

Bação, F., V. Lobo & M. Painho [2005]: *The self-organizing map, the Geo-SOM, and relevant variants for geosciences*. In: Computers & Geosciences, Vol. 31, No. 2, pp. 155 –163.

Balley, S., C. Parent & S. Spaccapietra [2004]: *Modeling Geographic Data with Multiple Representations*. In: International Journal of Geographical Information Science, Vol. 18, Issue 4, pp. 327-352.

Berners-Lee, T., J. Hendler & O. Lassila [2001]: *The Semantic Web*. Scientific American, May 2001.

Brocki, L. [2007]: *Kohonen Self-Organizing Map for the Traveling Salesperson Problem*. In: Recent Advances in Mechatronics 2007, pp. 116-119.

Brunner, K. [2003]: *Kartographische Reliefdarstellung - bewährte Methoden für das Printmedium.* In: W. G. Koch (Hrsg.): Theorie 2003. Vorträge der 8. Dresdener Sommerschule für Kartographie am 25./26. September 2003 an der TU Dresden. Kartographische Bausteine, Band 26, Dresden, S. 75-95.

Callier, S. & H. Saito [2011]: *Automatic Road Extraction from Printed Maps*. In: Proceedings of the IAPR Conference on Machine Vision Applications (IAPR MVA 2011), June 13-15, 2011, Nara, Japan.

Cao, R. & C. L. Tan [2002]: *Text/graphics separation in maps*. In: Proceedings of the Fourth International Workshop on Graphics Recognition Algorithms and Applications, pp. 167-177.

Carvajal, F., E. Crisanto, F.J. Aguilar, F. Aguera & M.A. Aguilar [2006]: *Greenhouses Detection Using an Artificial Neural Network with a Very High Resolution Satellite Image*. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVI, Part 2. Vienna (Austria), July 2006.

Chiang, Y.-Y., C. A. Knoblock, C. Shahabi & C.-C. Chen [2009]: *Automatic and Accurate Extraction of Road Intersections from Raster Maps*. In: GeoInformatica, June 2009, Vol. 13, Issue 2, pp. 121-157.

Coppini, G., R. Poli & G. Valli [1995]: *Recovery of the 3-D shape of the left ventricle from echocardiographic images*. In: IEEE Transactions on Medical Imaging 1995, Vol. 14, pp. 301-317.

Devadas, R., R. J. Denhama & M. Pringlea [2012]: *Support Vector Machine Classification of Object-based Data for Crop Mapping, using Multi-temporal Landsat Imagery*. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIX-B7.

Dhar, D.B. & B. Chanda [2006]: *Extraction and Recognition of Geographical Features from Paper Maps*. In: International Journal on Document Analysis and Recognition, Vol. 8, Issue 4, pp. 232–245.

Eklund, T., Back, B. & Vanharanta, H. [2002]: *Assessing the feasibility of self-organizing maps for data mining financial information*. In: Proceedings of the 10th European Conference onInformation Systems (ECIS), Vol. 1, pp. 528-537.

Esri [2012]: *ArcGIS – Hilfebibliothek. Desktop 10.0*. http://help.arcgis.com/de/arcgisdesktop/10.0/help/index.html#//00s900000004000000 (07.10.2012).

Esri [2013]: *ArcGIS – Hilfebibliothek. ArcGIS-Hilfe 10.1*. http://resources.arcgis.com/de/help/main/10.1/index.html#//000800000047000000 (09.11.2013).

Ester, M., H.-P. Kriegel, J. Sander, & X. Xu [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), pp. 226-231.

Felden, C. [2006a]: *Extraktion, Qualitätssicherung und Klassifikation unstrukturierter Daten*. In: HMD Praxis der Wirtschaftsinformatik 247, pp. 54–62.

Felden, C. [2006b]: *Text Mining als Anwendungsbereich von Business Intelligence*. In: Analytische Informationssysteme. Business Intelligence-Technologien und -Anwendungen. Springer-Verlag, pp. 283–304.

Fincke, T., V. Lobo & F. Bação [2008]: *Visualizing self-organizing maps with GIS*. In: Pebesma, E., Bishr, M. & Bartoschek, T. (Eds.): GI-Days 2008 - Proceedings of the 6th Geographic Information Days, Münster.

Frey C.W. [2012]: *Monitoring of complex industrial processes based on self-organizing maps and watershed transformations*. In: 2012 IEEE International Conference on Industrial Technology (ICIT), pp. 1041–1046.

Frischknecht, S., E. Kanani & A. Carosio [1998]: *Automatic Interpretation of Topographic Maps: A Raster-Based Approach*. In: Graphics Recognition Algorithms and Systems Lecture Notes in Computer Science, Vol. 1389, pp. 207-220.

Garcia, C. & J.A. Moreno [2005]: *An Efficient Heuristic for the Traveling Salesman Problem Based on a Growing SOM-like Algorithm*. In: Proceedings of the International Conference in Coimbra, Portugal, 2005, pp. 177-180.

Goodrich, M. T. & R. Tamassia [2001]: *Algorithm Design: Foundations, Analysis, and Internet Examples*. Wiley. 724 pages.

Graeff, B. & A. Carosio [2002]: *Automatic Interpretation of Raster-Based Topographic Maps by Means of Queries*. FIG XXII International Congress Washington, D. C., published on CD-ROM, 12 pages.

Guha, S., R. Rastogi & K.Shim [1998]: *CURE: An Efficient Clustering Algorithm for Large Databases*. In: SIGMOD '98 Proceedings of the 1998 ACM SIGMOD international conference on Management of data, pp. 73-84.

Hake, G., D. Grünreich & L. Meng [2002]: *Kartographie*. Walter de Gmbh Gruyter, Berlin, New York, 604 S.

Hai, T. & Y.-L. Bao [2008]: *Road Extraction from Color Raster Urban Traffic Map*. In: MMIT '08 Proceedings of the 2008 International Conference on Multi-Media and Information Technoligy, pp. 381-384.

Hanka, R., T.P. Harte, A.K. Dixon, D.J. Lomas & P.D. Britton [1996]: *Neural networks in the interpretation of contrast-enhanced magnetic resonance images of the breast*. In: Proceedings of Healthcare Computing. Harrogate: UK, 1996, pp. 275-283.

Heinzle, F. & M. Sester [2004]: *Derivation of Implicit Information from Spatial Data Sets with Data Mining*. In: 20th Congress of the International Society for Photogrammetry and Remote Sensing, Vol. 35, pp. 335-340.

Heinzle, F., K.-H. Anders & M. Sester [2007]: A*utomatic detection of pattern in road networks-methods and evaluation*. In: Proceeding of joint workshop visualization and exploration of geospatial data, Vol. XXXVI-4/W45.

Herold, H., G. Meinel, R. Hecht & E. Csaplovics [2012]: *A Geobia Approach to Map Interpretation Multitemporal Building Footprint Retrieval for High Resolution Monitoring Of Spatial Urban Dynamics*. In: Proceedings of the 4th GEOBIA, May 7-9, 2012 - Rio de Janeiro – Brazil, pp. 252.

Hertz, J., A. Krogh & R. G. Palmer [1991]: *Introduction to the Theory of Neural Computation*. Westview Press; First Paperback Edition (June 24, 1991), 350 pages.

Hinneburg, A. & D.A. Keim [1998]: *An Efficient Approach to Clustering in Multimedia Databases with Noise*. In: Proceedings of the 4th International Conference on Knowledge Discovery and Datamining (KDD‘98), New York, 1998, pp. 58-65.

Huang. Y.H. [2013]: *webmagic*. http://www.oschina.net/p/webmagic (10.10.2013).

Ifeachor, E.C. & K.G. Rosen [1994]: *The Development of an Expert System for the Analysis of Umbilical Cord Blood at Delivery*. In: Proceedings of the International Conference on Neural Networks and Expert Systems in Medicine and Healthcare. Plymouth: UK, 1994, pp. 394-402.

Imhof, E. [1965]: *Kartographische Geländedarstellung*. Berlin, de Gruyter, 425 S.

Innocent, P.R., M. Barnes & R. John [1997]: *Application of the fuzzy ART/MAP and MinMax/MAP neural network models to radiographic image classification*. Artificial Intelligence in Medicine 1997, Vol. 11, No. 3, pp. 241-263.

Jain, N,, S. Sharma & R.M. Sairam [2013]: *Content Base Image Retrieval using Combination of Color, Shape and Texture Features*. In: International Journal of Advanced Computer Research (ISSN (print): 2249-7277, ISSN (online): 2277-7970) Vol. 3, No. 1, Issue 8.

Jardin, P. & Séverin, E. [2011]: *Predicting corporate bankruptcy using a self-organizing map: An empirical study to improve the forecasting horizon of a financial failure model.* In: Decision Support Systems, Vol. 51, Issue 3, pp. 701-711.

Jones, P., J. L. Williams & S. Lannon [2000]: *Planning for a Sustainable City: an Energy and Environmental Prediction Model*. In: Journal of Environmental Planning and Management, Vol. 43, No. 6, pp. 855-872(18).

Karkanis, S., G.D. Magoulas, M. Grigoriadou & M. Schurr [1999]: *Detecting abnormalities in colonoscopic images by textural description and neural networks*. In: Proceedings of Work. On Mach. Learn. in Med. Appl., Advance Course in Artif. Intell.-ACAI99. pp. 59-62.

Karkanis, S., G.D. Magoulas & N. Theofanous [2000]: *Image recognition and neuronal networks: intelligent systems for the improvement of imaging information*. In: Minimal Invasive Therapy and Allied Technologies, Vol. 9, No.3-4, pp. 225-230.

Kaufman, L. & P.J. Rousseeuw [1990]: *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 1990, S. 342.

Khotanzad, A. & E. Zink [2003]: *Contour line and geographic feature extraction from USGS color topographical paper maps*. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 1, pp. 18-31.

Khurana, D. & S. Kumar [2012a]: *An Improved Approach for Caption Based Image Web Crawler*. In: IJCSMS International Journal of Computer Science & Management Studies, Vol. 12, Issue 02, ISSN (Online): 2231–5268.

Khurana, D. & S. Kumar [2012b]: *An Improved Approach to perform Crawling and avoid Duplicate Web Pages*. In: IJCSMS International Journal of Computer Science & Management Studies, Vol. 12, ISSN (Online): 2231–5268.

Kleinberg, J. & E. Tardos [2006]: *Algorithm Design*. Addison Wesley, pp. 92–94.

Kohonen, T. [2001]: *Self-Organizing Maps*. Springer Series in Information Sciences, Vol. 30, 3rd ed. 2001, 502 pages.

Kou, M.-M. & Q.-Z. Wang & T.-D. Tan [2012]: *Recognition and Extraction of Road from Color Digitial Raster Graphic*. In: Computer Engineering, Vol.38, No. 13, pp. .

Krause, C. [1993]: *Kreditwürdigkeitsprüfung mit neuronalen Netzen*. Düsseldorf: Idw-Verlag 1993, S. 37.

Lampinen, J. & T. Kostiainen [2000]: *Self-Organizing Map in Data-Analysis-Notes on Overfitting and Overinterpretation*. In: ESANN'2000 proceedings – European Symposium on Artificial Neural Networks, pp. 239-244.

Lannon, S. C., D. K. Alexander & P.J. Jones [2007]: *Housing stock surveys in GIS systems using pattern recognition*. In: Winstanley, A. C. ed. GISRUK 2007 - Proceedings of the Geographical Information Science Research UK Conference. Maynooth, Co. Kildare: NUI Maynooth, pp. 430-434.

Lanza, A., D. Malerba, F.A. Lisi, A. Appice & M. Ceci [2002]: *Generating Logic Descriptions for the Automated Interpretation of Topographic Maps*. In: Graphics Recognition: Algorithms and Applications, Lecture Notes in Computer Science, Vol. 2390, pp. 200-210.

Li, W.-J., H.-S. Ru, T.-J. Zhao & W.-M. Zang [2009]: *A New Algorithm of Topical Crawler*. In: Computer Science and Engineering. WCSE '09. Second International Workshop on 28-30 Oct. 2009, Vol. 1, pp. 443–446.

Li, X. & C. Claramunt [2006]: *A Spatial Entropy-Based Decision Tree for Classification of Geographical Information*. In: Transactions in GIS, Vol. 10, Issue 3, pp. 451-467.

Lichodzijewski, P., A. N. Zincir-heywood & M. I. Heywood [2002]: *Host-Based Intrusion Detection Using Self-Organizing Maps*. In: proceedings of the IEEE International Joint Conference on Neural Networks, pp. 1714-1719.

Liu, W., K.C. Seto, & Z. Sun [2005]: *Urbanization prediction with an ART-MMAP neural network based spatiotemporal data mining method*. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVI, Part 8/W27, pp. 1682-1777.

Liu, H., E. Milios & L. Korba [2008]: *Exploiting Multiple Features with MEMMs for Focused Web Crawling*. In: Proceeding NLDB '08 Proceedings of the 13th international conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems, pp. 99 – 110.

Lüscher, P., R. Weibel & A. Mackaness [2008]: *Where is the terraced house?: on the use of ontologies for recognition of urban concepts in cartographic databases*. In: Ruas, A. Headway in spatial data handling. Berlin, DE, pp. 449-466.

Malerba, D., A. Appice, A. Varlaro & A. Lanza [2005]: *Spatial Clustering of Structured Objects*. In: Inductive Logic Programming: ILP 2005, Lecture Notes in Artificial Intelligence, Vol. 3625, pp. 227-245.

Meinel, G., H. Herold & R. Hecht [2006]: *Automatische Ableitung siedlungsstruktureller Grundlagendaten auf Basis digitaler Bildverarbeitung, GIS und räumlicher Statistik*. In: Strobl, Blaschke, Griesebner (eds.), Angewandte Geoinformatik 2006, Beiträge zum 18. AGIT-Symposium Salzburg, S. 423-429.

Mishra, S., A. Jain & A.K. Sachan [2010]: *Smart Approach to Reduce the Web Crawling Traffic of Existing System using HTML based Update File at Web Server*. In: International Journal of Computer Applications, Vol. 11, No.7, pp. 34.

Narayan Das, N. & E. Kumar [2012]: *Hidden Web Query Technique for Extracting the Data from Deep Web Data Base*. In: Proceedings of the World Congress on Engineering and Computer Science, Vol. I WCECS 2012, ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online).

Ng, R. & J. Han [1994]: *Efficient and Effective Clustering Method for Spatial Data Mining*. In: Proceedings of Int'l Conf. on Very Large Data Bases, Santiago, Chile, 1994, pp. 144-155.

Nuernberger, A. & Detyniecki, M. [2006]: *Externally growing self-organizing map and its application to e-mail database visualization and exploration*. In: Applied Soft Computing, Vol. 6, Issue 4, pp. 357–371.

Pao, Y.-H. [1989]: Adaptive Pattern Recognition and Neural Networks; Addison-Wesley Verlag Reading, 1989.

Phee, S.J., W.S. Ng, I.M. Chen, F. Seow-Choen & B.L. Davies [1998]: *Automation of colonoscopy Part II: Visual-control aspects*. In: IEEE Engineering in Medicine and Biology Magazine, Vol. 17, Issue 3, pp. 81-88.

Pütz C. & E.J. Sinz [2010]: *Modellgetriebene Ableitung von BPMN-Workflowschemata aus SOM-Geschäftsprozessmodellen*. In: Modellierung 2010. LNI 161 GI 2010. Köllen, Bonn, S. 253-268.

Rowe, N. C. [2002]: Marie-4: *A High-Recall, Self-Improving Web Crawler That Finds Images Using Captions*. In: IEEE Intelligent Systems, Vol. 17, pp. 8-14.

Ruas, A. & J.P. Lagrange [1995]: *Data and Knowledge Modeling for Generalization*. In: Muller, J-C., Lagrange, J.-P., and Weibel, R. (eds.): GIS and Generalization: Methodological and Practice, pp. 73-90.

Pant, G., P. Srinivasan & F. Menczer [2004]: *Crawling the Web*. In: Web Dynamics: Adapting to Change in Content, Size, Topology and Use. Springer-Verlag, pp. 153-178.

Rojas, Raúl [1993]: Theorie der neuronalen Netze : Eine systematische Einführung. Berlin: Springer, 446 S.

Salton, G. & M. J. McGill [1983]: *Introduction to modern information retrieval*. 448 pages.

Sander, J., E. Martin, H.-P. Kriegel & X. Xu [1998]: *Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications*. In: Data Mining and Knowledge Discovery, Vol. 2, Issue 2, pp. 169–194.

Sester, M. [1995]: *Lernen struktureller Modelle für die Bildanalyse*. Dissertationen, Deutsche Geodätische Kommission (DGK), Reihe C, Nr. 441, 116 S.

Sester, M. [2000]: *Maßstababhängige Darstellungen in digitalen räumlichen Datenbeständen*. Habilitation, Fakultät für Bauingenieur- und Vermessungswesen, Universität Stuttgart, Deutsche Geodätische Kommission (DGK), Reihe C, Nr. 544, 108 S.

Sester, M. [2007]: *Generierung von Kartographischen Präsentationen in Maßstäben 1:25.000 und 1:50.000 mit PUSH und TYPIFY*. In: Mitteilungen des Bundesamtes für Kartographie und Geodäsie, Verlag des Bundesamtes für Kartographie und Geodäsie (BKG), 2007.

Schleinkofer, M.-F. [2007]: *Wissensbasierte Unterstützung zur Erstellung von Produktmodellen im Baubestand*. Dissertation, Technische Universität München, 135 S.

Spielman, S. E. & J.-C. Thill [2008]: *Social area analysis, data mining, and GIS*. In: Computers, Environment and Urban Systems, Vol. 32, Issue 2, pp. 110–122.

Steinhauer, J.H., T. Wiese, C. Freksa & T. Barkowsky [2001]: *Recognition of Abstract Regions in Cartographic Maps.* In: Proceedings of the International Conference on Spatial Information Theory: Foundations of Geographic Information Science, pp. 306 – 321.

Veropoulos, K., C. Campbell & G. Learmonth [1998]: *Image processing and neural computing used in the diagnosis of tuberculosis*. In: Colloq. Intelligent Meth. in Health. and Med. Appl. York: UK.

Viglino, J-M. & M. Pierrot-Deseilligny [2003]: *A Vector Approach for Automatic Interpretation of the French Cadatral Map*. In: Proceeding of the Seventh International Conference on Document Analysis and Recognition (ICDAR '03), pp. 304-309.

Walter, V. [2007]: *Automatic urbanity cluster detection in street vector databases with a raster-based algorithm*. In: Proceedings of the 23nd international cartographic conference (ICC): Cartography for everyone and for you, 4-10 August 2007 Moscow, Russia, 10 pages, published on CD-ROM.

Walter, V. & F. Luo [2011]: *Automatic interpretation of digital maps*. In: ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 66, Issue 4, pp. 519–528.

Wang, W., J. Yang & R. Muntz [1997]: *STING: A Statistical Information Grid Approach to Spatial Data Mining*. In: Proceedings of 23rd VLDB, Athens, Greece, 1997, pp. 186-195.

Wang, X. & H. J. Hamilton [2003]: *DBRS: A Density-Based Spatial Clustering Method with Random Sampling*. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2003, pp. 563–575.

Whigham P. A. [2005]: *Local Modelling by SOM partitioning and linear regression for Ecological Modelling*. In Zerger, A. and Argent, R.M. (eds) MODSIM 2005 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, December 2005, pp. 1319-1325.

Zell, A. [1997]: *Simulation Neuronaler Netze*. Oldenbourg Wissenschaftsverlag.

Zhang, T., R. Ramakrishna & M. Livny [1996]: *BIRCH: An Efficient Data Clustering Method For Very Large Databases*. SIGMOD Record, Vol. 25, No. 2, pp. 103-114.

Zhang, X., J.E. Stoter & T-H. Ai [2008]: *Formalization and automatic interpretation of map requirements*. In: Proceedings of the 11th ICA workshop on generalisation and multiple representations, 20-21 June 2008, Montpellier. 12 pages.

Zhu, Y. & H. Yan [1997]: *Computerized tumor boundary detection using a Hopfield neural network.* In: IEEE Transactions on Medical Imaging 1997, Vol. 16, Issue 1, pp. 55-67.

Lebenslauf

Persönliche Angaben

Fen Luo
Geboren am 20.05.1979 in Guangxi, VR China

Schulausbildung

1985 – 1991	Grundschule Zhaoping, VR China
1991 – 1997	Gymnasium Zhaoping, VR China

Studium

1997 – 2001	Bachelorstudium der Geoinformatik und Kartographie an der Universität Wuhan, VR China
2001 – 2006	Diplomstudium der Geodäsie und Geoinformatik an der Universität Stuttgart

Beruf

2006 – 2006	Wissenschaftliche Mitarbeiterin am Institut für Photogrammetrie der Universität Stuttgart
2006 – 2007	Application Engineer bei der Firma PRO DV
2007 – 2010	Software Developer bei der Firma ISB AG
2009 –	Promotion zum Dr.-Ing. an der Universität Stuttgart (Externe Promotion)
2010 –	Softwareentwicklerin bei der Firma ICON Systemhaus