Skript ***Grundlagen Geoinformationssysteme***

Dr. Peter Menzel

Sommersemester 2021

Das folgende Skript basiert auf den Vorlesungsunterlagen für die Lehrveranstaltung *Grundlagen Geoinformationssysteme* an der TUBAF aus dem Sommersemester 2019. Die Autoren waren Prof. H. Schäben und Prof. C. Gerhards.

Inhaltsverzeichnis

1 Grundlegende Einführung und Klärung der Begriffe

Geo-...

Der Silbe "geo-" stammt aus dem Griechischen und bedeutet "die Erde betreffend". So ist die ursprüngliche Bedeutung von "Geometrie" (*geo-metria*) die "Lehre von der Messbarkeit/Vermessung der Erde".

Informatik ist die Wissenschaft über die systematische Verarbeitung von Informationen, insbesondere die automatisierte Verarbeitung mittels Computern (siehe *Computerwissenschaften*). Sie befasst sich mit der Entwicklung und Anwendung von Informationstechnologien.

Geoinformatik@TUBAF Geoinformatik im Sinne der TU Bergakademie Freiberg befasst sich mit der Anwendung von Methoden der Informatik (aber auch der Mathematik) auf geowissenschaftliche Probleme. Dabei wird der gesamte Komplex der Geowissenschaften erfasst. Die geowissenschaftlichen Datenmenge ist in der letzten Jahrzehnten, vor allem durch verbesserte Messmethodik und der Möglichkeit globaler Messungen über Satelliten, exponentiell gewachsen. Es werden daher systematische Methoden benötigt, um diese umfassende Wissensbasis zu verwalten und auszuwerten.

Relevante Technologien und Ansätze umfassen dabei das formale Studium der Suche und Nutzung von geologischen Informationen, die Analyse und Speicherung von Wissen, Informationen, sowie das Erkennen von Anwendern und die Analyse des Anwenderverhaltens. Web-basierte geowissenschaftliche Angebote, Data Mining und künstliche Intelligenz sowie elektronische Veröffentlichung und Präsentation von Informationen sind dabei einige praktische Anwendungen.

Geomatik Geomatik befasst sich mit der Erforschung von systematischen Ansätzen, um räumliche Daten im Zuge von wissenschaftlichen, administrativen, rechtlichen oder technischen Tätigkeiten zu sammeln und zu bearbeiten. Levinson (*Language and space*, 1992; Seite 72) beschreibt Geomatik als die "Kunst", Wissenschaft und Technologie, geografisch referenzierte Informationen auszuwerten und zu bearbeiten.

Daten, Informationen und Wissen

Nach [Bil10] sind **Daten**, im klassischen Sinn der Informatik, Zeichen, welche durch einen Computer gespeichert und verarbeitet werden können. Im erweiterten Sinn fallen auch ganze Bilder, Texte, Graphiken und Symbole unter diesen Begriff. Daten werden verwendet, um die quantitativen und qualitativen Eigenschaften von Objekten und Sachverhalten des aktuellen Interesses zu beschreiben bzw. diese Beschreibung zu ermöglichen. Sie sind in erster Linie Computer-interpretierbar. Für den Menschen haben sie aber ohne bekannte Regeln für Interpretation und Struktur kaum Aussagekraft. Ein thematisch zusammengehörige Datenmenge wird auch als *Datensatz* bezeichnet. Jedes einzelne Element dieser Menge ist ein *Datum*.

Im Gegensatz zu Daten entsteht **Information** aus der Anwendung von Regeln und Anweisungen auf Daten. Es handelt sich um das Ergebnis der Anwendung von Operationen wie Transformationen, Regeln und Wissen auf Daten durch einen Anwender, welcher sowohl mit den Daten als auch mit den Operationen vertraut ist. Dadurch werden neue Fakten und interpretierbare Ergebnisse unter den gegebenen Voraussetzungen erzeugt. Informationen sind immer auch an ein Informationsmittel (mündl. Sprache, Schrift, Abbildungen) gebunden, welches eine Kommunikation ermöglicht. Die Kenntnis der Datenorganisation und des Datenzwecks erlaubt es dem Anwender, mit den Daten im Sinne von interpretierbaren Informationen zu arbeiten. Information teilt sich in drei Ebenen auf:

1. **Syntax**: interne Codierung, Struktur und Repräsentation der Information der zugrunde liegenden Daten;

2. **Semantik**: Bedeutungskontext der zugrunde liegenden Daten;

3. **Pragmatik** und **Kommunikation**: Verwendung und Wiedergabe der zugrunde liegenden Daten.

Das **Wissen** eines Wissensträgers umfasst die Menge aller von ihm als wahr angenommen Aussagen über einen repräsentierten Sachverhalt, welche tatsächlich wahr sind. Es umfasst damit sowohl die Daten selbst, als auch die daraus abgeleiteten Informationen, die Verknüfung zu anderen Daten und Informationen sowie die Kenntnis der verwendeten Methoden und der einer Anwendung zugrunde liegenden Motivation.

System, Informationssystem und Datenbanken

Ein **System** ist eine gegliederte Menge an Objekten. Diese Menge ist durch eine so genannte Systemgrenze nach Außen abgeschlossen, steht mit diesem Äußenäber in einer bestimmten Beziehung. Die einzelnen Teile (Elemente, Objekte) eines System können selbst wiederum Systeme sein (Subsysteme) und in Beziehung zueinander stehen. Durch die Systemgrenze ist klar definiert, welche Objekte Teil des Systems sind und welche nicht. Im Sinne der Informationstechnologie ist ein System ein zu einem bestimmten Zweck eingesetzte Kombinationen aus Hard- und Softwarekomponenten, welche in einer Weise wechselwirken, dass sie als eine aufgaben-, sinn- oder zweckgebundene Einheit angesehen werden können.

Ein so genanntes **Informationssystem** dient der rechnergestützten Erfassung, Verarbeitung, Analyse und Kommunikation von Daten und Informationen. Es besteht aus Hard- und Softwarekomponenten, Daten und deren Anwendungen. Sie stellen ein Informationsangebot aufgrund einer nutzerbasierten Informationsnachfrage bereit.

Ein **Datenbanksystem** (DBS) ist ein System zur systematischen elektronischen Datenverwaltung, das es erlaubt, große Datenmengen effizient, eindeutig und dauerhaft zu speichern. Es stellt dem Anwender Teilemengen aus den Daten bedarfsgerecht zu Verfügung. Dein DBS besteht aus zwei Komponenten, einem Datenbankmanagementsystem (DBMS) und den verwalteten Daten, welche in einer Datenbank (DB) strukturiert gespeichert sind. Das DBMS organisiert intern die Speicherung und kontrolliert alle lesenden und schreibenden Zugriffe auf die Datenbank. Zur Abfrage und Verwaltung der Daten durch einen Anwender wird eine vom DBMS unterstützte Datenbanksprache (z. B. SQL) verwendet.

Raumbezug

Ein Raumbezug besteht immer dann, wenn Daten oder Informationen räumlich in Beziehung gesetzt werden. Dabei wird zwischen direkten und indirektem Raumbezug unterschieden ([Cre15]). Ein **direkter Raumbezug** (*primäre Metrik*) wird zum Beispiel in den Geowissenschaften (Geologie, Geophysik, usw.) und bestimmten Ingenieurwissenschaften (z.B. Vermessungswesen) verwendet. Hier wird der Raumbezug über die Angabe zwei- oder dreidimensionalen Koordinaten für jedes Datum hergestellt. Diesen Koordinaten liegt ein definiertes *Koordinatenreferenzsystem* zugrunde. Die primäre Metrik erlaubt einerseits die Angabe von Genauigkeiten und anderseits die Berechnung von z.B. Abständen zwischen zwei Datenlokationen. Verschiedene primäre Metriken lassen sich durch bekannten Transformationen leicht ineinander überführen. In anderen Disziplinen, z.B. Statistik oder Soziologie, beruht der **(indirekte) Raumbezug** auf der Angabe von qualitativen Größen, welche einer schwächeren, *sekundären Metrik*, folgen. Hier ist die Bestimmung von räumlichen Genauigkeiten und die Abstandsberechnung nur schwerlich möglich. Beispiele für sekundäre Metriken sind z.B.

- *Kennziffern* für räumliche Gliederungsbereiche (z.B. Postleitzahlen, Flurstücksnummern)
- *Namen*, die einen Ort benennen und ein räumliches Gebiet umschreiben (z.B. Ortsnamen, Ländernamen, Lagebezeichnungen)
- *Adressen* (Stadt, Straßenname, Hausnummer etc.) als Basis für amtliche Datenerhebungen.

Aufgrund der qualitativen Natur dieser Metriken lassen sie sich nur schwer systematisch ineinander oder in primäre Metriken umwandeln. Primäre Metriken können dagegen z.B. durch Klassifikation leicht in sekundäre Metriken umgewandelt werden.

Georeferenzierung bezeichnet im engeren Sinne die Zuweisung eines Raumbezugs zu einem Datensatz. Hierbei wird angegeben, in welchem Raumbezug der Datensatz vorliegen soll, unabhängig davon, ob er zum Zeitpunkt der Zuweisung bereits in diesem Raumbezug vorliegt.

So gilt zum Beispiel eine Luftbildaufnahme (Rasterbild) bereits als georeferenziert, wenn für einige Pixel zusätzlich die Koordinaten in einem bekannten Koordinatensystem (z.B. das GPS-Koordinatensystem WGS 1984) zugewiesen wurden. Der Vorgang der **Geocodierung** beschreibt allgemein die Überführung von einem Raumbezug in einen Anderen, also zum Beispiel die Transformation aus einem lokalen Koordinatensystem in ein globales Koordinatensystem oder die Zuweisung von Postleitzahlen basierend auf der bekannten Position. In der Praxis wird heutzutage nicht mehr exakt zwischen Georeferenzierung und Geocodierung unterschieden. Im Allgemeinen wird heute unter dem Begriff *Georeferenzierung*, vor allem im Kontex von GIS-Software, sowohl die Zuweisung des Raumbezugs als auch die simultane Transformation in diesem Raumbezug verstanden.

Im Rahmen von Geoinformationssystemen wird sowohl mit primären als auch mit sekundären Metriken gearbeitet. Dabei weisen die einzelnen Daten immer mindestens einen direkten Raumbezug und gegebenen Falls mehrere indirekte Raumbezüge auf.

Geo(-grafische) Informations Systeme - GIS

Geoinformationssysteme ermöglichen die digitale Verwaltung und Bearbeitung von räumlichen - georeferenzierten - Daten und Informationen. Dabei bedeutet georeferenziert, dass diese Daten auf einen Punkt oder eine Region auf der Erdoberfläche bezogen werden können (siehe *Raumbezug*). Die räumlichen Daten können aus den verschiedensten Quellen bezogen und dann in einer GIS Datenbank gespeichert und verwaltet werden. Ein GIS ermöglicht dabei die Bearbeitung und Transformation dieser Daten, um relevante Informationen zu extrahieren und die Daten zu neuen Daten zu kombinieren, auf deren Basis dann Entscheidungen getroffen werden können und die es ermöglichen, räumlichen Beziehungen der verschiedenen Eingabedaten zu verstehen ([BC94]).

Ein anschauliches Beispiel für eine aktuelle web-basierte GIS-Anwendung finden Sie hier: Coronavirus COVID-19 Global Cases by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University.

Definition nach Bill (2010, [Bil10]):

> Ein Geo-Informationssystem (GIS) ist ein rechnergestütztes System, das aus Hardware, Software und Daten besteht und mit dem sich raumbezogene Problemstellungen in unterschiedlichen Anwendungsgebieten modellieren und bearbeiten lassen. Die dafür benötigten Daten / Informationen können digital erfasst und redigiert, verwaltet und reorganisiert, analysiert sowie alphanumerisch und graphisch präsentiert werden. GIS bezeichnet sowohl eine Technologie, Produkte als auch Vorhaben zur Bereitstellung und Behandlung von Geoinformationen.

Beispiel: Ein digitales Höhenmodell (*Digital Elevation Model* - DEM)

Ein digitales Höhenmodell, das auf einer ausreichenden großen Informationsbasis beruht, erlaubt es, in einer vordefinierten Region jedem beliebigen Punkt numerisch eine Höhe zuzuweisen und die Höheninformation als Geoobjekt zu repräsentieren. Dieses umfasst zum Beispiel:

- die Daten als eine Menge von Tupeln (*x, y, Höhe*)
- Metadaten, wie physikalische Einheiten, geografisches Referenzsystem, Aufnahmezeitpunkt, Datenunsicherheit, Name des Autors ...
- zusätzliches Wissen und Modellannahmen
- Möglichkeiten und Verfahren zur räumliche Interpolation / Approximation / Vorhersage
- digitale Repräsentation (Datenmodell, Datenstruktur)
- Visualisierung

Die Höhe eines Gebietes ist ein komplexes Phänomen aus der realen Welt. Um es in einer Computer-verarbeitbaren Form zu repräsentieren, ist ein Abstraktionsprozess notwendig. Dieser um fasst zum Beispiel die Vereinfachung mittels Diskretisierung, dabei wird der kontinuierliche Raum in eine endliche Menge von Elementen zerlegt, welchen dann die Höheninformationen zugeordnet werden. Für ein digitales Höhenmodell könnten folgende Repräsentationen verwendet werden:

Rastermodell: Das Untersuchungsgebiet wird in ein regelmäßiges Gitter aus gleichförmigen Zellen zerlegt. Jede Zelle erhält einen Höhenwert. Wenn das Ordnungsschema bekannt ist, lässt sich dieses Gitter leicht als eine aufeinanderfolgende Liste aus Höhenwerten reduzieren und speichern. Zu Interpretation dieser Liste ist die Kenntnis des Ordnungsschema zwingend notwendig. Digitale Bilder werden bevorzugt im Rastermodell repräsentiert.

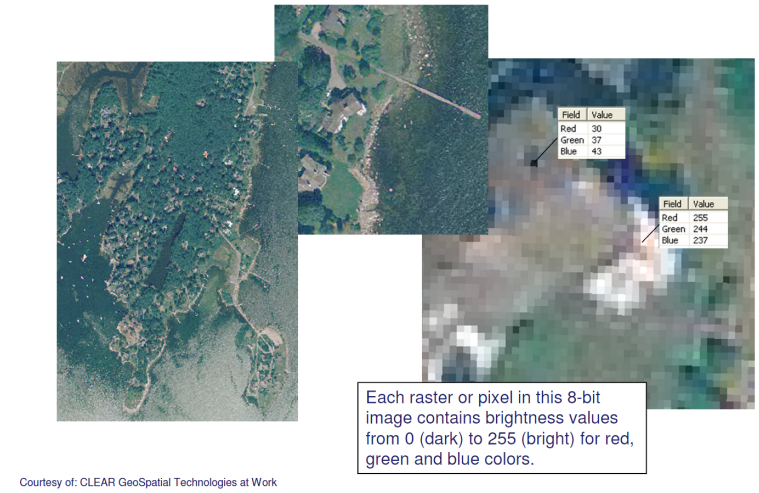


Abbildung 1: Digitales Luftbild als Beispiel für Daten im Rastermodell.

Vektormodell: Die Informationen werden zum Beispiel über Punkte, Konturlinien (Linien gleicher Höhe) oder Polygone gespeichert. Für jeden Punkt sind dabei mindestens die Koordinaten und der zugeordnete Höhenwert bekannt. Die Punktpositionen müssen dabei keinem Ordnungsschema folgen und können beliebig im Untersuchungsgebiet verteilt sein. Linien und Polygone verknüpfen die Punktinformationen und können selbst wieder eigene Eigenschaften besitzen. Punkte, Linien und Polygone lassen sich als leicht interpretierbare Tabellen digital speichern. Die Vektorinformationen können dabei unabhängig von einander oder über einen **Graph** verknüpft sein. Dies ist zum Beispiel bei einer **Delaunay Triangulation** der Fall, welche die gegebenen Punktdaten zu zusammenhängenden Dreiecken (Polygone) verknüpft. Ein solcher Graph ermöglicht es zum Beispiel, Nachbarschaftsbeziehungen zwischen Punkten oder Polygonen effizient abzuleiten. Das Vektormodell erlaubt die sehr effiziente Repräsentation beliebiger zwei- und dreidimensionaler geowissenschaftlicher Strukturen. Die dabei verwendeten Datenmodelle und -strukturen sind aber sehr viel komplexer als beim Rastermodell.

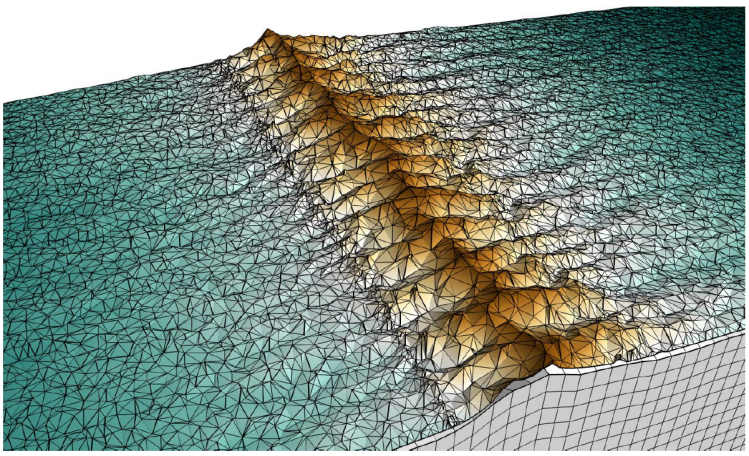


Abbildung 2: Repräsentation eines Höhenmodells über eine Menge verknüpfter Dreiecke.

2 Geoinformationssysteme - Definition, Funktionen und Anwendungen

Definition nach Bill (2010, [Bil10]):

> Ein Geo-Informationssystem (GIS) ist ein rechnergestütztes System, das aus Hardware, Software und Daten besteht und mit dem sich raumbezogene Problemstellungen in unterschiedlichen Anwendungsgebieten modellieren und bearbeiten lassen. Die dafür benötigten Daten / Informationen können digital erfasst und redigiert, verwaltet und reorganisiert, analysiert sowie alphanumerisch und graphisch präsentiert werden. GIS bezeichnet sowohl eine Technologie, Produkte als auch Vorhaben zur Bereitstellung und Behandlung von Geoinformationen.

2.1 Das Vier-Komponenten-Modell eines GIS

Im engsten Sinn ist ein Geoinformationssystem nach Bill (2010, [Bil10]) ein Computersystem aus Hardware und Software zum Erfassen, Speichern, Bearbeiten und Darstellen von räumlichen Informationen. Dabei wird zwischen Funktion und Struktur eines GIS unterschieden. Diese lassen sich jeweils in vier Hauptkomponenten unterteilen (**Vier-Komponenten-Modell**).

Die wichtigsten funktionalen Komponenten werden über das Akronym
IMAP - **Input** (Eingabe), **Management** (Verwaltung), **Analysis** (Auswertung/Analyse), **Präsentation** (Ausgabe/Darstellung)
zusammen gefasst. Die Eingabe umfasst dabei sowohl das automatische und manuelle Erfassen von Rohdaten und die Fähigkeit zur Integration von sekundären Daten in das Softwaresystem sowie die Georeferenzierung und Transformation in den Raumbezug des aktuellen Projekts. Die Datenverwaltung umfasst einerseits den Aufbau der zugehörigen Datenbank inklusive der Schnittstellen für Datenzugriff und Dateneingabe, andererseits die Bearbeitung der vorhandenen Geoobjekte, sowie die Erstellung neuer Objekte. Für die Datenauswertung werden die Daten nach spezifischen Kriterien aus der Datenbank abgefragt (*query*), kombiniert und analysiert. Es werden Modelle aufgebaut, welche den zu untersuchenden Sachverhalt abbilden. Basierend auf diesen Modellen sollen Entscheidungen getroffen werden. Die Primärdaten und die erzeugten sekundären Daten können auf verschiedenste Art visualisiert und präsentiert werden und als Datenprodukte (Ausgabedaten) für die weitere Verwendung exportiert werden.

Zusätzlich zu den vier funktionalen Komponenten lässt sich der Aufbau eines GIS über vier strukturelle Hauptkomponenten beschreiben:

1. Hardware
2. Software
3. Daten
4. Mensch

Die Hardware umfasst alle physischen Komponenten, die notwendig sind, um die die beschriebenen Funktionalitäten ausführen zu können. Die Software umfasst neben den verschiedenen Benutzerschnittstellen (GUI) zur Dateneingabe, -ausgabe, -präsentation und -bearbeitung, das zur Datenverwaltung notwendige DBMS und sämtliche Operationen und Methoden, die zur Bearbeitung und Auswertung notwendig sind. Unter Daten werden sämtliche eingegeben, erstellten, abgeleiteten und ausgegebenen Daten und Geoobjekte zusammenfasst (siehe *Daten und Objekte aus Sicht eines GIS*). Die strukturelle Komponente Mensch"bezeichnet sowohl den operativen Anwender, der über das GIS Daten erstellt und zu Datenprodukten zusammenführt, als auch den Nutzer dieser Datenprodukte, der über die GIS-Datenbank auf diese Produkte zugreift, um sie zu analysieren und Entscheidungen zu treffen. Diese strukturellen Komponenten interagieren untereinander und bilden so ein System oder Netzwerk.

2.2 Daten und Objekte aus Sicht eines GIS

Aus Sicht eines GIS lassen sich Daten nach ihrem Ursprung oder ihrer Quelle in **Primärdaten** und **Sekundärdaten** unterscheiden. Primärdaten sind alle Daten, welche direkt in der Natur

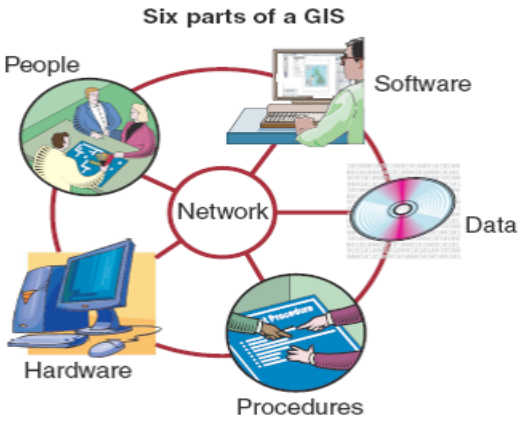


Abbildung 3: Schematische Darstellung der strukturellen GIS-Komponenten

beobachtet, d.h. gemessen wurden. Dazu zählen zum Beispiel Messdaten aus Geophysik oder Vermessungswesen, geologische Feldkartierungen, geochemische oder hydrologische Messungen und Satellitenrohdaten, sowie Luftbilder. Diese Daten sind in erster Linie Eingangsdaten für GIS-Projekte. Sekundärdaten werden dagegen durch Bearbeitung und Interpretation von Primärdaten erzeugt. Dazu zählen z.B. topografische Karten oder digitale Geländemodelle, Karten der Bouguer-Schwere/-Anomalie oder Landnutzungsklassifizierungen. Ausgabedaten oder Datenprodukte aus GIS-Projekten sind immer Sekundärdaten. Sekundärdaten können aber auch als Eingangsdaten für GIS-Projekte verwendet werden. Über den Vorgang der Digitalisierung lassen sich neue Geodaten basierend auf existierendem Kartenmaterial aufnehmen. Analoge Karten werden dafür zuerst eingescannt. Durch den Scanvorgang liegt jetzt ein digitales Rasterbild vor, das zwar einen lokalen Raumbezug hat (Pixelkoordinaten), aber noch nicht im Raumbezug des GIS-Projektes vorliegt. Über so genannte Passpunkte im Bild, an denen neben den Pixelkoordinaten auch Koordinaten im Projekt-Koordinatensystem bekannt sind, kann das Rasterbild in das Koordinatensystem des Projektes transformiert werden. Dieser Vorgang wird im Abschnitt *Koordinatensysteme, Kartenprojektion und Koordinatentransformation* näher erläutert. Neue Geoobjekte werden dann auf Basis dieser georeferenzierten Rasterkarten manuell erstellt, in dem zum Beispiel für im Bild erkennbare Punktobjekte neue Punktobjekte in einem neuen Vektor-Layer erzeugt werden.

Zusätzlich lassen sich Daten nach ihrer Funktion in **klassische Daten**, **Metadaten** und **Herkunftsdaten** (*data provenance / linage*) unterteilen. Klassische Daten beschreiben die gemessenen oder interpretierten Eigenschaften eines realen Sachverhaltes und umfassen sowohl Primär- als auch Sekundärdaten. Metadaten sind "Daten über Daten"([GAMR15]) und beschreiben zum Beispiel Ort, Zeit und Art der Messung, das verwendete Koordinatensystem usw. Datensätze ohne aussagekräftige Metadaten lassen sich durch einen Anwender nur schwer verwenden und sind somit quasi nutzlos. Herkunftsdaten sind ein spezieller Typ von Metadaten und beziehen sich vor allem auf Sekundärdaten. Sie erlauben es, die Ëntstehungsgeschichteëines Datenproduktes abzuleiten. Sie umfassen u.a.

- die verwendeten Methoden und Algorithmen,
- die Reihenfolge der Bearbeitungsschritte,
- die verwendete Hard- und Software,
- die verwendeten Eingangsdaten,
- den oder die Autoren des Produktes.

Herkunftsdaten dienen der Daten-Transparenz und erlauben u.a. die Reproduzierbarkeit eines Produktes. Herkunftsdaten werden aktuell nur sporadisch und inkonsistent erhoben. Spezielle Datenbanksysteme und Datenversionierung sind Gegenstand aktueller Forschung.

> Data should be documented adequately enough to find it, interpret it, and understands its provenance. (Seite 58; NSF, 2009)

In einem klassischen GIS wird des weiteren grundsätzlich zwischen *Geodaten* und *Sachdaten* unterschieden. Geodaten weisen immer einen Raumbezug auf und können sowohl im Vektor-, als auch im Rastermodell vorliegen. Sie lassen sich so auch grafisch darstellen. Diese beiden Datenmodelle wurden bereits grundsätzlich eingeführt (siehe Einführung) und werden unter dem Thema *Datenmodellierung* später ausführlich behandelt. Nach Bill (2010, [Bil10]) können Geodaten in *Geometriedaten* und so genannte *topologische Daten* unterschieden werden. Geometriedaten beschreiben die Form und Lage von einzelnen Geoobjekten, wohingegen topologische Daten die räumlichen Zusammenhänge und Nachbarschaftsbeziehungen zwischen einzelnen Geoobjekten und ihren Teilobjekten beschreiben. **Sachdaten** liegen initial ohne Raumbezug vor. Sie werden auch als thematische Daten, Attribute oder beschreibende Daten bezeichnet. Sie repräsentieren nicht-räumliche Elemente wie Texte, Tabellen, Nummern, Namen und Eigenschaften. Sie können an Geodaten / Geoobjekte gekoppelt werden und erhalten so einen Raumbezug. In erster Linie werden Sachdaten als Datenbanktabellen gespeichert, welche es erlauben, über Anfragen und relationale Algebra darauf zuzugreifen. Ein spezieller Typ von Sachdaten sind **grafische Attribute** wie Objektfarbe und -füllung, Textgröße und -farbe, Linienstärke, Symbolart, -größe und -rotation. usw. Sie dienen allein der grafischen Darstellung von Daten und Objekten und können wiederum auf vorhandenen Geo- und Sachdaten beruhen. Werden grafische Attribute geändert, ändert sich zwar die Darstellung eines Objektes, nicht aber die mit diesem Objekt verknüpften Daten.

Häufig werden Daten in einem GIS in sogenannten **Geoobjekten** gruppiert. Diese sind zusammenhängende, räumlich abgeschlossene Einheiten mit ähnlichen thematischen Eigenschaften oder gleicher oder ähnlicher Bedeutung. Geoobjekte stellen Modelle realer Sachverhalte dar. So kann zum Beispiel ein einfacher Punkt jeden realen Sachverhalt abbilden, bei dem die räumliche Ausdehnung unwichtig und nur die Position relevant ist. Anhand ihrer Ausdehnung lassen Sie sich in

- **Punktobjekte** (Dimension = 0),
- **Linienobjekte** (Dimension = 1),
- **Flächenobjekte** (Dimension = 2) und
- **Volumenobjekte** (Dimension = 3)

unterteilen.

Punktobjekte Bei Punktobjekten handelt es sich um unabhängige räumliche verteilte Objekte oder Ereignisse, z. B. Bohrungen, Messpunkte oder Einzellokationen. Im Vektormodell besitzen Punktobjekte zwar eine Punktkoordinate, aber keinerlei Ausdehnung. Im Rastermodell handelt es sich um ein unabhängiges Einzelpixel mit der Ausdehnung eines Pixels. Im oben gezeigten Beispiel ist das Punktobjekt ein einzelner Baum, dargestellt über ein "Baumsymbolän einer gegebenen Position. Die Symbolausdehnung ist ein rein grafisches Attribut und sagt nichts über die Ausdehnung des Punktobjektes aus.

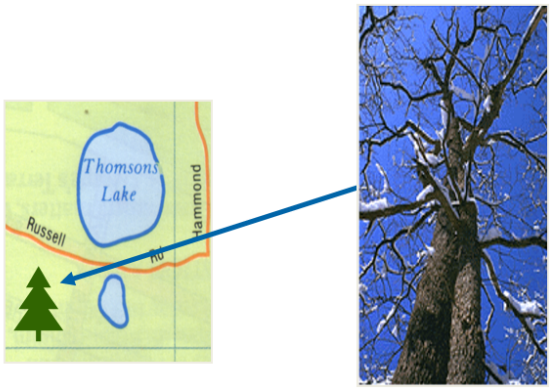


Abbildung 4: Punktobjekte.

Linienobjekte Linienobjekte sind Objekte, welche eine Länge aufweisen, deren Breite aber nicht notwendigerweise relevant ist. Im Vektormodell handelt es sich um eine Sequenz miteinander über Liniensegmente verbundener Punktlokationen ohne eine definierte "Breite". Im Rastermodell handelt es sich um aufeinander folgende Pixel, die Breite entspricht einer Pixelbreite. Im oben gezeigten Beispiel handelt es sich um eine Straße, dargestellt als Linie. Die dargestellte Linienbreite entspricht nicht notwendigerweise der realen"Breite der Straße.

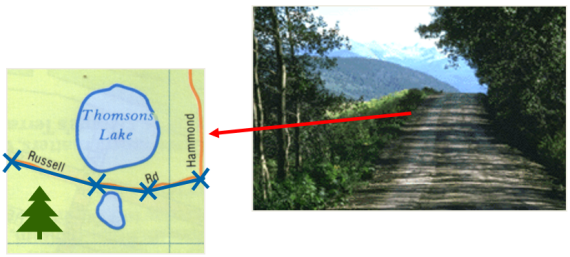


Abbildung 5: Linienobjekte.

Flächenobjekte Flächenobjekte sind Objekte mit einer zweidimensionalen räumlichen Ausdehnung. Sie sind räumlich begrenzt. Man kann ihre Fläche und z.B. die Länge der Grenze berechnen. Im Vektormodell werden Flächenobjekte über Polygone beschrieben. Diese Polygone sind wiederum durch einen geschlossenen Ring aus Liniensegmenten aufgebaut, welcher die Polygongrenze darstellt. Im Rastermodell besteht ein Flächenobjekt aus einer Menge verbundener Pixel mit gleichen/ähnlichen Attributen. Im obigen Beispiel wird ein See über ein Polygon mit einer Grenze aus Liniensegmenten dargestellt.

Abbildung 6: Flächenobjekte.

Volumenobjekte Dreidimensionale geschlossene Einheiten werden als Volumenobjekte bezeichnet und zumeist entweder über die geschlossene Grenzfläche (Vektormodell) oder zusammenhängende Voxel (3D Pixel, Volumen-Pixel; Rastermodell) repräsentiert. Im klassischen GIS sind dreidimensionale Objekte eher selten, es wird sich zumeist auf zweidimensionale Objekte beschränkt.

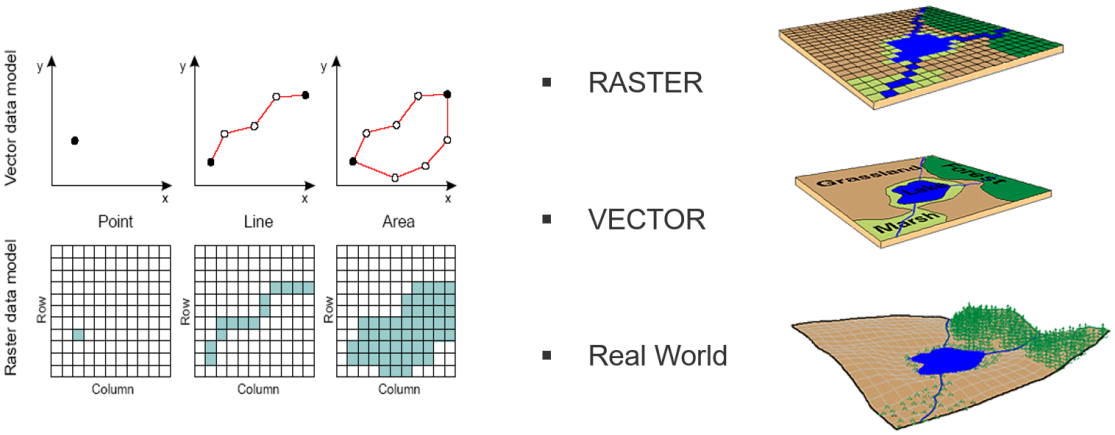


Abbildung 7: Verschiedene Objekte und ihre Repräsentation im GIS.

2.3 Grundlegende Funktionen eines GIS

GIS wird in der geowissenschaftlich Praxis hauptsächlich eingesetzt, um verschiedenen Datensätze miteinander zu kombinieren und digitale Karten zu erstellen. In erster Linie ist das Ziel, räumliche Daten und die zugehörigen Meta-Daten so zu verarbeiten, dass räumliche Sachverhalte beschrieben oder charakterisiert, verstanden und vorhergesagt werden können. Dies wird durch

- Organisation,
- Visualisierung,
- räumliche Abfragen,
- Kombination,
- Analyse,
- Modellierung und
- Simulation

der Geodaten erreicht und erfolgt zu meist durch einen Ëxperten", welcher über das notwendige Wissen verfügt. Zumeist sollen basierend auf den erzielten Ergebnissen Entscheidungen getroffen werden.

Ein GIS-Projekt unterteilt sich dabei in drei Abschnitte:

1. **Aufbau der räumlichen Datenbank**, alle verfügbaren Eingabedaten werden in einer Datenbank zusammengeführt;
2. **Datenverarbeitung**, Herausarbeiten und Ableiten der für das Projekt relevanten räumlichen Strukturen;
3. **Datenintegration und -modellierung**, Kombination der verschiedenen Daten, um Erkenntnisse zu gewinnen.

Bonham-Carter (1994, [BC94]) beschreibt die grundlegenden Anwendungen innerhalb eines GIS. Diese überlappen sich zum Teil bezüglich der notwendigen Funktionalitäten. Im Folgenden werden diese GIS-Basisanwendungen näher erläutert.

Dateneingabe und -aufnahme

Verschiedene vorhandenen Datensätze können sehr unterschiedliche Datenstrukturen aufweisen und sollen dennoch möglichst einheitlich in einer GIS Datenbank verwaltet werden können. Dafür werden so genannte Datenmodelle, z.B. Raster oder Vektormodell, verwendet, um die Daten zu beschreiben, zu speichern, auszuwerten und zu bearbeiten. Je nach vorliegendem Datenmodell und zugrunde liegender Datenstruktur lassen sich bestimmte Operationen besonders effizient auf diesen Daten ausführen. Darauf wird unter dem Thema *Datenmodellierung* näher eingegangen.

Innerhalb eines GIS werden zusammengehörige Daten als so genannte Ebenen (*Layer*) gruppiert. Jede Ebene kann als thematische Karteängesehen werden. Alle Ebenen liegen im gleichen Raumbezug vor und können einander überlagern. Dabei spielt die Reihenfolge der verschiedenen Ebenen eine entscheidende Rolle. Hintergrundinformationen wie z.B. ein Luftbild oder ein DEM werden thematischen Daten wie z.B. Linien eines Straßennetzes oder Punktlokationen von Messstellen so überlagert, dass beide Datensätze gemeinsam ausgewertet werden können.

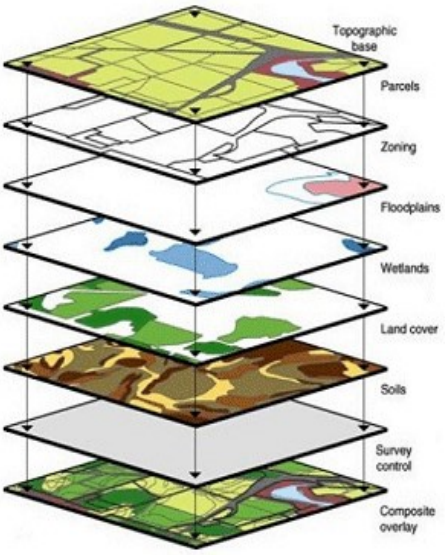


Abbildung 8: Verschiedene Ebenen in einem GIS. Die Pfeile geben die Richtung der Überdeckung an.

Visualisierung

Da das menschliche Wahrnehmungsvermögen komplexe räumliche Beziehungen visuell (Bilder, Karten) sehr viel besser erfassen kann, als über eine reine Zahlen- und Zeichendarstellung (Texte, Tabellen), ist eine der wichtigsten Anwendungen in einem GIS, alle gesammelten Daten grafisch so aufzubereiten, dass eine schnelle und umfassende optische Analyse durchgeführt werden kann. So werden z.B. geochemische Messpunkte in einem Gebiet als farbige Symbole dargestellt. Die Position jedes Symbols entspricht der Position eines Messpunkte und die Farbe codiert den Messwert basierend auf einer vorgegebenen Farbtabelle. Da die Darstellungsparadigmen zwischen den verschiedenen wissenschaftlichen Disziplinen sich zum Teil sehr stark unterscheiden, müssen innerhalb einer GIS-Software viele verschiedenen Darstellungsmethoden vorgehalten werden. Die erstellten Kartendarstellungen können zudem als exportier- und druckbare Kartenprodukte inklusive der notwendigen Meta-Daten bereit gestellt werden.

Zusätzlich zur räumlichen Darstellung der Geodaten ist es in einem GIS häufig möglich, die Sachdaten auch ohne räumlichen Bezug zu visualisieren und statistisch auszuwerten (Histogramme, Boxplots, usw.).

Räumliche Abfragen

Über Visualisierung lassen sich zwar räumliche Strukturen leicht identifizieren, spezifische Informationen, z.B. welchen Wert für ein Attribut eine Struktur genau aufweist, sind damit aber nur schwer abzuleiten. Diese sind aber für eine tiefgehende Analyse der Daten essentiell. Um diese spezifischen Informationen ableiten zu können, werden die räumlichen Daten mit den Sachdaten über Datenbankabfragen verknüpft. Es gibt zwei grundsätzlich unterschiedliche räumliche Abfragen:

1. Welche Charakteristiken weist eine gegebene Position auf?
2. An welchen Positionen tritt eine gegebene Charakteristik auf oder welche Geoobjekte weisen eine gegebene Charakteristik auf?

Eine *Charakteristik* meint in diesem Fall spezifische Werte für eine gegebene Kombination aus Attributen.

Die erste Abfragekategorie befasst sich vor allem mit der interaktiven Abfrage. Der Anwender markiert eine Lokation, ein Gebiet oder ein einzelnes Geoobjekt auf dem Bildschirm und das GIS ermittelt alle Werte der damit verknüpften Attribute/Sachdaten. Diese werden entweder direkt am Bildschirm ausgegeben (z.B. über eine MessageBox) oder es werden die betreffenden Einträge in der zugehörigen Attributtabelle markiert (siehe A in der folgenden Abbildung).

Die zweite Kategorie entspricht eher klassischen allgemeinen Datenbankabfragen. Es wird eine Anfrage an die GIS-Datenbank bezüglich bestimmter Attribute/Sachdaten gestellt, z.B.

```
Attribut X = Wert
```

oder

```
Attribut Y > Wert_1 UND Attribut Z < Wert_2 .
```

Das GIS ermittelt alle Einträge, für die die Anfrage positiv ist und markiert in der Karte die mit den positiven Einträgen verknüpften Lokationen, Gebiete oder Geoobjekte (siehe B in der folgenden Abbildung).

Zusätzlich lassen sich beide Kategorien verknüpft verwenden, z. B. indem nach allen Lokationen, Gebieten oder Geoobjekten gefragte wird, welche die gleiche Attributkombination wie ein interaktiv markiertes Objekt aufweisen. Ein weiterer spezieller Abfragetyp sind so genannte **topologische Abfragen**, bei denen nach räumlichen Beziehungen gefragt wird. Es kann z.B. nach allen Nachbarobjekten zu einem markierten Objekt gefragt werden oder nach allen Objekte, welche sich weiter als 5 km von einem definierten Objekt entfernt befinden. Diese Informationen liegen häufig nicht explizit vor, sondern müssen aus der Datenstruktur der betreffenden Geoobjekte/Geodaten extrahiert oder berechnet werden.

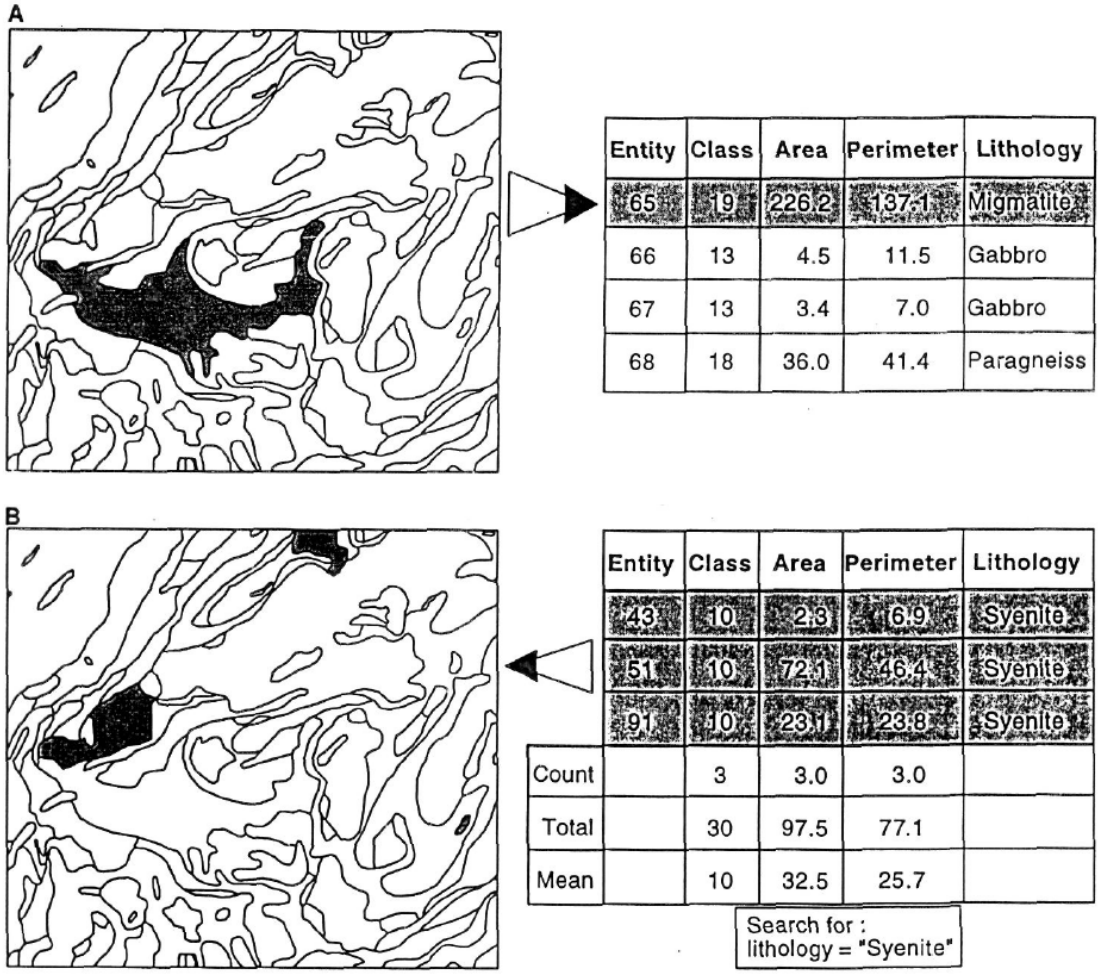


FIG. 5-14. Interactive spatial query of a geological map. **A**. Identifying the attributes of a polygon *selected on a map view*, as indicated in an associated polygon attribute table. **B**. Identifying those polygons on the map that have the attribute called lithology equal to "syenite", as *selected from the polygon attribute table*.

Abbildung 9: Räumliche Abfragen nach Bonham-Carter (1994, [BC94]).

Datenkombination

Ein GIS erlaubt es, die verschiedenen in der GIS-Datenbank verfügbaren Datensätze zu kombinieren. Dieser Vorgang wird auch als Datenintegration bezeichnet. Datenkombination führt zu einem besseren Verständnis und Interpretation der räumlichen Sachverhalte. So kann man z.B. eine geologische Karte mit einer Visualisierung von geochemischen Probenlokationen / Analyseergebnissen überlagern, um ein besseren Überblick über die chemischen Eigenschaften / Komponenten der einzelnen geologischen Einheiten in einem Gebiet zu bekommen.

Datenkombination kann auf verschiedenen Art und Weise erfolgen:

1) Datenebene Bei der Datenkombination auf der Ebene der Daten werden mittels mathematischer Methoden und Modelle neue Attribute aus einer Kombination bereits bekannter Attribute berechnet. Dadurch werden die Sachdaten eines Datensatzes um das neu erstellte Attribut erweitert. So kann zum Beispiel das Verhältnis zweier chemischer Elemente

```
ratio_Pb_Zn = Pb_content / Zn_content
```

von größerem geochemischen Interesse sein, als die einzelnen gemessenen Elementgehalte. `ratio_Pb_Zn` ist dabei das neue Attribut für das Verhältnis Blei zu Zink, `Pb_content` und `Zn_content` die bereits bekannten Attribute für die Blei- und Zink-Gehalte.

2) Karten- oder Layerebene Jedes GIS-Layer kann als einzelne, thematische"Karte angesehen werden. Diese Karten können über die Mittel der **Map-Algebra** zu neuen GIS-Layern zusammengeführt werden. Dadurch wird ein neues Layer und damit ein neuer GIS-Datensatz als Kombination der Eingangsebenen erzeugt. Die Eingangsdaten bleiben dabei unverändert. Angenommen, es liegen 2 GIS-Layer als Eingangsdaten vor, eine Landnutzungsklassifikation basierend auf einem Luftbild und eine Karte des Nitratgehaltes im Boden. Es soll jetzt ein neues Layer generiert werden, welches die Nitratbelastung im Ackerboden darstellt. Über die Mittel der Map-Algebra wird dem neuen Layer an allen Positionen, in denen die Landnutzungsklasse "Acker" im ersten Eingangsdatensatz auftritt, der Wert für den Nitratgehalt aus dem zweiten Eingangslayer gespeichert. Für alle anderen Landnutzungsklassen (z.B. "Wald", "Wiese", "Bebauung") wird dagegen ein default-Wert gespeichert . Dieser könnte in diesem Fall z.B. "nan" oder "-99999" sein. Der Wert "0" ist in diesem Fall ungeeignet, das er ein valider Wert für die Nitratkonzentration ist und so eine Unterscheidung zw. den Acker- und Nicht-Ackerbereichen schwerer zu erkennen ist.

3) Visualisierungsebene Datenkombination kann auch rein visuell erfolgen, indem verschiedenen Layer gemeinsam überlagert visualisiert werden. Verschiedenen Vektorlayer lassen sich sehr leicht überlagern, wohingegen sich verschiedenen Rasterlayer ganz oder teilweise so verdecken, dass nur das "oberste" Layer zu sehen ist. Dem kann man durch den Einsatz von Layer-Transparenz entgegenwirken, bei der bestimmte Layer teilweise "durchsichtig" dargestellt werden und so die darunter liegenden Layer durchscheinen. In diesem Fall werden keine neuen Daten oder Layer erstellt und die visualisierten Layer auch nicht verändert, sondern nur verschiedenen Objekte aus verschiedenen Layern gemeinsam dargestellt.

Der Prozess der Kombination verschiedener Karten zu neuen Karten wird auch als "map modelling" oder "cartographic modelling" bezeichnet ([BC94]).

Datenanalyse

Datenanalyse beschreibt den Prozess, "Bedeutung" oder "aussagekräftige Information" aus gegebenen Daten abzuleiten, um darauf aufbauend Entscheidungen treffen zu können. Dies kann sowohl im räumlichen als auch im nicht-räumlichen Sinne erfolgen.

Nicht-räumliche Datenanalyse erfolgt über statische Auswertung der vorhandenen Sachdaten ohne räumlichen Bezug, z.B. über visuelle Datenauswertung anhand von statistischen Grafiken (Histogramme, Box-Plots), der Berechnung von statistischen Größen oder dem Anpassen von statistischen Modellen an die Daten.

Die räumliche Auswertung erfolgt über die Visualisierung, Datenkombination, räumlichen Abfragen, sowie über räumliche Statistik (Geostatistik). Räumliche Statistik erlaubt die Berechnung von statistischen räumlichen Zusammenhängen. Ihre Grundannahme ist dabei, dass Daten, welche

sich nah sind, sich immer ähnlicher sind, als Daten, welche sich weiter von einander entfernt befinden. Diese räumliche Korrelation kann dann dazu verwendet werden, Datenwerte an unbekannten Lokationen zu schätzen (siehe *Kriging*; [Cre15]).

Vorhersagemodelle

Eine der wichtigsten Anwendungen eines GIS ist die Vorhersage von Sachverhalten basierend auf den bekannten Daten in der GIS-Datenbank. Diese Vorhersage basiert unter anderem auf der Datenanalyse und dient ebenfalls zur Entscheidungsfindung.

Es kann zwischen nicht-räumlicher und räumlicher Vorhersage unterschieden werden. Nicht-räumliche Vorhersage basiert auf klassischen mathematischen und/oder statistischen Modellen (z.B. *Regression*), bei denen aus der kombinierten Auswertung von Attributkombinationen auf ein unbekanntes Attribut geschlossen wird, ohne dass die Position der Daten in Betracht gezogen wird. So lässt sich ein Modell für ein Attribut `Baugrundstabilität` z. B. als Funktion der Attribute `Bodenfeuchte`, `Untergrundkompaktion` und `Untergrundlithologie` ausdrücken und vorhersagen.

Bei der räumlichen Vorhersage ist das Ziel, ein an wenigen Lokationen bekanntes Attribut an Positionen vorherzusagen, an denen es initial unbekannt ist. Diese räumliche Vorhersage wird auch als Interpolation bezeichnet. Mathematisch lassen sich die meisten Interpolationsverfahren als Summe über die gewichteten n bekannten Datenwerte ausdrücken mit

$$\hat{f}(\vec{x}) = \sum_{i=1}^{n} w_i \cdot f(\vec{x}_i).$$

$\hat{f}(\vec{x})$ ist der vorherzusagende Wert an einer Stelle $\vec{x}$, w_i das Gewicht für den bekannten Wert $f(\vec{x}_i)$ an einer Stelle $\vec{x}_i$.

Es wird zwischen **deterministischer** und **statistischer** Interpolation unterschieden. Bei der deterministischen Interpolation erfolgt die Vorhersage basierend auf einem feststehenden mathematischen Modell ohne Betrachtung der tatsächlichen räumlichen Korrelation der bekannten Daten. Ein Beispielverfahren ist hier *IDW* (*inverse distance weighting*, inverse Distanzwichtung), welches den vorherzusagenden Wert als gewichtetes Mittel der bekannten Werte berechnet. Die Gewichte werden über den inversen Abstand zu den bekannten Datenpunkten bestimmt. Statistische Interpolationsverfahren, wie zum Beispiel *Kriging*, berücksichtigen bei der Bestimmung der Interpolationsgewichte die räumliche Korrelation. Eine nähere Betrachtung dieses Thema erfolgt unter dem Themenkomplex *Räumliche Vorhersage / Interpolation* im weiteren Verlauf dieser Lehrveranstaltung.

Eine sehr häufige Anwendung für Interpolation ist die Übertragung von Werten für ein Attribut, das nur an Einzelpunkten vorliegen, auf die Fläche (Raster, Polygone) zur Erzeugung von Kartendarstellungen für dieses Attribut (siehe nachfolgende Abbildung).

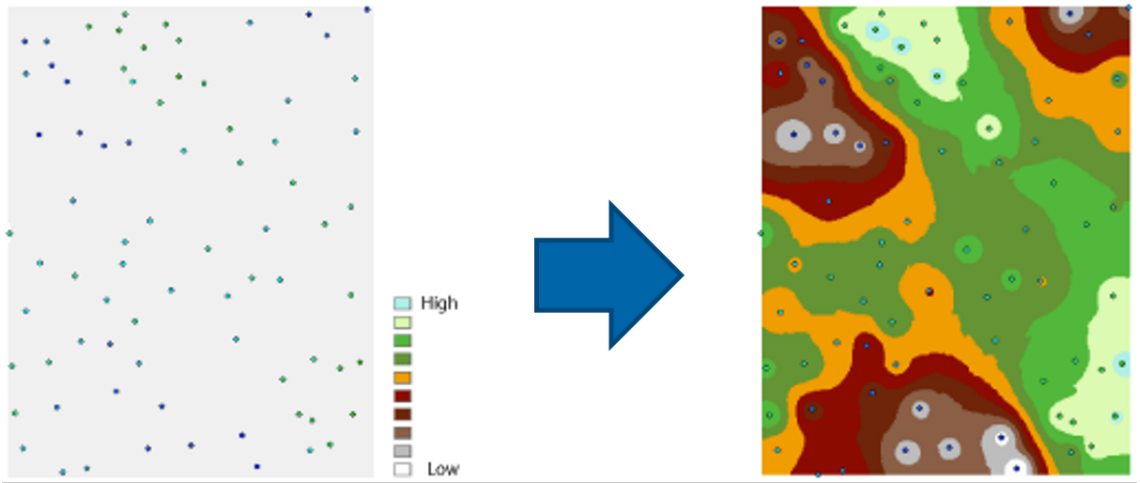


Abbildung 10: Interpolation zur Übertragung von Punktinformationen in die Fläche.

2.4 Beispiele für praktische GIS-Anwendungen

GIS kann sowohl für wissenschaftliche, soziale, politische und kommerzielle Untersuchungen und Planungsvorhaben eingesetzt werden, sofern räumliche Daten die Grundlage bilden.

In den Geowissenschaften sind die Hauptanwendungen in erster Linie die Analyse von räumlichen Strukturen und die räumliche Vorhersage von relevanten Attributen. Diese Vorhersagen kann dann zum Beispiel genutzt werden, um Hochwasser-gefährdete Regionen an Küsten und Binnengewässern zu identifizieren oder Standorte für Bebauung zu planen. Auch lässt sich zum Beispiel dass ökonomische Potential für die Rohstoffgewinnung in einer Region ermitteln. In dem Umweltwissenschaften kann die Verbreitung bedrohter Spezies oder Ökosysteme analysiert und so zum Beispiel Schutzgebiete ausgewiesen werden. Über GIS lässt sich aber auch die räumliche Verteilung von sozio-ökonomischen, ethnischen oder politischen Faktoren innerhalb einer Gesellschaft analysieren und in politischen oder kulturellen Entscheidungen berücksichtigen.

Ein anschauliches Beispiel für eine soziologische und politische Anwendung ist das so genannte "Gerrymandering", also der räumliche Zuschnitt der Wahlbezirke in den USA. Dieser basiert auf der Analyse der gesellschaftlichen Struktur hinsichtlich politischer Ausrichtung und sozio-ökonomischen Faktoren, mit dem sich u.a. auch Wahlergebnisse manipulieren lassen. In der folgenden Abbildung wird gezeigt, wie sich durch die Änderung des Zuschnitts der Wahlbezirke eine eigentlich klare Wählersituation (überwiegend blau) zu einem überwiegend roten Ergebnis verändern lassen könnte. Ursprünglich war und ist das Überarbeiten der Wahlbezirksgrenzen basierend auf statistischen Erhebungen der Bevölkerung als Mittel der Gleichbehandlung verschiedener Wählergruppen gedacht. Es wird in Form des "Gerrymandering"jedoch zunehmend wahltaktisch von den politischen Akteuren missbraucht.

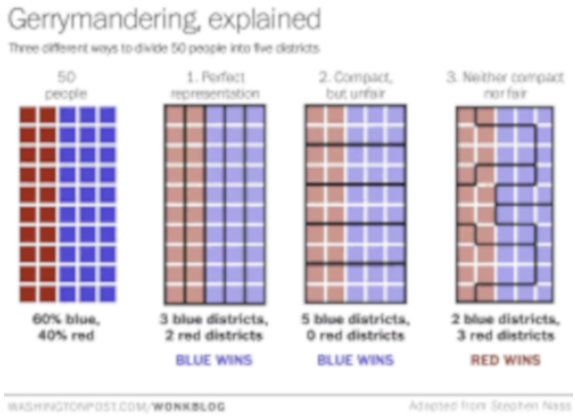


Abbildung 11: Gerrymandering zur Beeinflussung des Wahlergebnisses.

Eine tiefere Erläuterung finden sie unter https://de.wikipedia.org/wiki/Gerrymandering oder in diesem Video.

3 Kartenprojektion, Koordinatensysteme und Koordinatentransformation

3.1 Koordinatensysteme und Kartenprojektion

Figur der Erde

Die reale Geometrie der Erdoberfläche ist sehr unregelmäßig und lässt sich mathematisch kaum beschreiben. Der Begriff "Figur der Erde" bezeichnet eine Approximation dieser Geometrie durch einfachere geometrische Grundformen, welche mathematisch beschreibbar sind. Sie entsprechen dabei aber immer noch den physikalischen Gesetzmäßigkeiten und lassen sich auf lokale Gegebenheiten beziehen. Diese Bezugs- oder Referenzflächen sind mathematisch, physikalisch oder über Festpunktfelder bestimmte Flächen, auf die sich Lagekoordinaten, Höhen oder Schwerepotentiale von Punkten beziehen.

Die Erde ist ein annähernd kugelförmiges Objekt im Raum, welches aufgrund seiner Rotation an den Polen leicht abgeplattet ist. Das bedeutet, ihre Oberfläche ist überall leicht gekrümmt. Diese Krümmung ist jedoch in jeder einzelnen Punktlokation vergleichsweise gering. Lokal begrenzte, kleinräumige Gebiete auf der Erde können als "flach" angesehen und über eine Ebene beschrieben werden. Dies ist jedoch nur dann möglich, wenn die Krümmung über dem Gebiet vernachlässigbar klein ist. Für größere Gebiete kann die Erdoberfläche als Kugelfläche approximiert werden. Der Radius dieser Erdkugel entspricht grob $R = 6370$ km. Für physikalische oder geodätische Anwen-

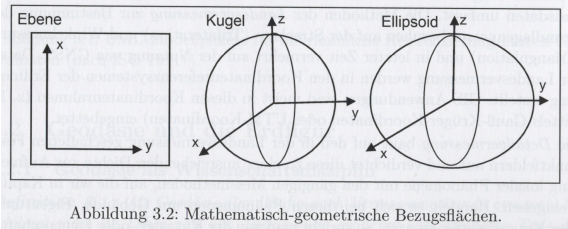


Abbildung 12: Mathematische Bezugsflächen nach Bill (2010; [Bil10]).

dungen reicht diese Approximation jedoch oft nicht aus. Die hierfür verwendete Bezugsfläche ist ein so genanntes **Rotationsellipsoid**. Dieses basiert auf einer **Meridianellipse**, welche um die Erdachse rotiert wird. Diese Ellipse lässt sich mathematisch durch ihre "große Halbachse" a (maximaler Radius vom Erdzentrum zum Äquator) und ihre "kleine Halbachse" b (minimaler Radius vom Erdzentrum zu den Polen) beschreiben. Die Abplattung f der Meridianellipse und damit des Referenzellipsoides lässt sich dann über $f = \frac{a-b}{b}$ bestimmen. Auch die Verwendung von Bezugsellipsoiden ist nur eine Annäherung an die Form der Erde. Es wurden und werden Ellipsoide mit verschiedenen Parametern verwendet, siehe die nachfolgende Tabelle (entnommen aus [Bil10]). Für viele, vor allem satelliten-basierte Anwendungen (z.B. GPS) wird aktuell das mittlere Ellipsoid des "World Geodetic System"(WGS'84) verwendet.

Tabelle 1: Verschiedene Referenzellispoiden nach [Bil10].

Name	**Entstehungsdatum**	a **in m**	b **in m**
Bessel	1841	6 377 397	6 356 079
Heyford	1924	6 378 388	6 356 912
Krassowskij	1940	6 378 245	6 356 863
GRS80≈WGS'84	1979 / 84	6 378 137	6 356 752

Neben der rein mathematisch beschriebenen Erdfigur lässt sich die Bezugsfläche auch aus physikalischen Annahmen ableiten. Ein Beispiel hierfür ist das so genannte **Geoid**. Es beschreibt eine Äquipotentialfläche des Schwerefeldes. Aufgrund einer ungleichen Massenverteilung im Untergrund ist diese Fläche gleicher Schwerkraft nicht überall gleich weit vom Zentrum der Erde entfernt. In der nachfolgenden Abbildung sehen Sie ein Modell (EIGEN-6C4) des Geoides, basierend u.a. auf aktuellen Satellitenmissionen.

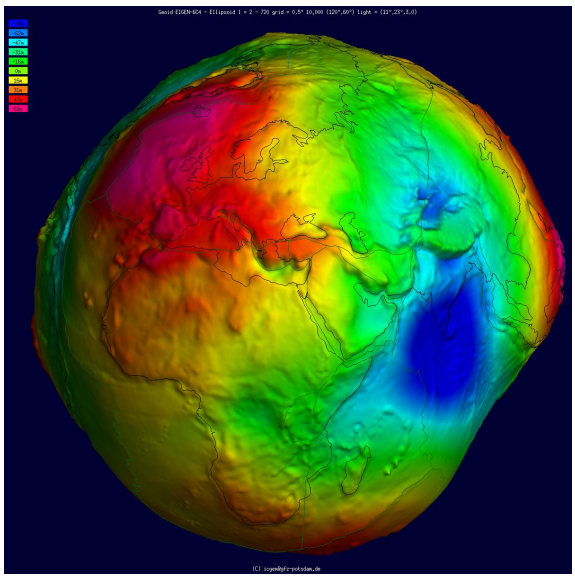


Abbildung 13: Physikalische Bezugsfigur Geoid - "Potsdamer Kartoffel". Die Darstellung ist stark überhöht (Faktor ≈1:10 000, http://icgem.gfz-potsdam.de).

Koordinatensysteme

Um Punktlokationen auf den verschiedenen Referenzflächen bestimmen zu können, wird eine primäre Metrik benötigt. Diese ist meist definiert über ein Koordinatensystem, bei dem jedem Punkt für jede Koordinatenrichtung ein Wert zugewiesen wird. Diese Werte werden als die Koordinaten dieses Punktes bezeichnet.

Die Bestimmung einer Punktposition im Raum ist grundsätzlich ein dreidimensionales Problem. Jeder Punkt weist hier drei Werte für die drei Koordinatenrichtungen auf. Dieses wird in der klassischen Landesvermessung und in den meisten GIS in ein zweidimensionales Lageproblem und ein eindimensionales Höhenproblem aufgespalten. Dies führt dazu, dass die "Höhe" in den meisten GIS nur als zusätzliches Attribut geführt wird und nicht in die Geometrieinformationen einfließt. Diese Trennung ist jedoch nicht mehr zeitgemäß. In vielen aktuellen GIS wird deshalb zumindest teilweise mit echten 3D Koordinaten gearbeitet.

Koordinaten auf der Kugel - geografische Koordinaten Ein Punkt auf der Erdoberfläche als Oberfläche eines beliebigen Rotationsellipsoides lässt sich über die beiden Winkel λ (Länge) und ϕ (Breite) beschreiben. λ und ϕ sind so genannte **sphärische Koordinaten**, da sich über die beiden Winkel jeder Punkt auf Kugelflächen beschreiben lässt. Eine Linie von Nord- zu Südpol durch einen beliebigen Punkt P auf der Erdoberfläche wird als **Meridian** bezeichnet. Die Länge (als der Winkel $\lambda(P)$) beschreibt den Winkel zwischen dem Meridian, der durch P verläuft und dem Zentral- oder Nullmeridian ($\lambda = 0$) auf der Äquatorialebene der Kugel. Der Nullmeridian verläuft per Definition durch Greenwich bei London. Die Breite (als der Winkel $\phi(P)$) beschreibt den Winkel zwischen P und dem Äquator ($\phi = 0$) entlang des Meridians durch P. λ verläuft zwischen -180° und 180° und ϕ zwischen -90° am Südpol und 90° am Nordpol.

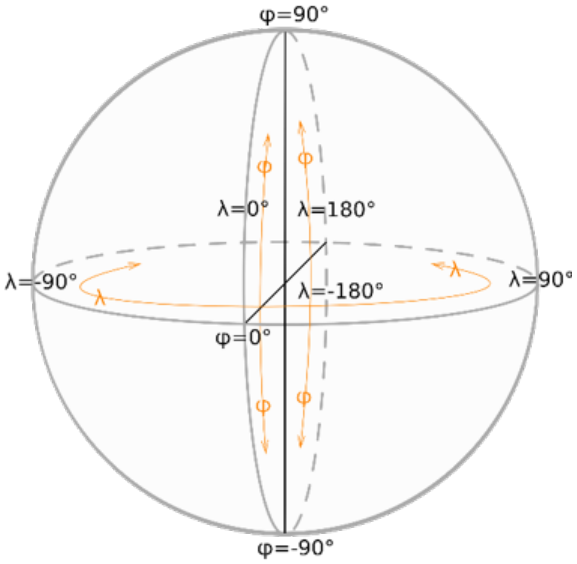


Abbildung 14: Definition der geografischen Koordinaten (https://de.wikipedia.org/wiki/Geographische_Koordinaten).

Ein so genannter **Großkreis** ist eine Linie auf der Kugel, welche durch den Schnitt einer Ebene durch das Kugelzentrum mit der Kugeloberfläche entsteht. Verläuft die Schnittebene nicht durch das Zentrum ist die Schnittlinie mit der Oberfläche ein so genannter **Kleinkreis**. Alle Meridiane (Längenkreise; d.h. Kreise gleicher Länge λ) und der Äquator sind Großkreise. Der Äquator ist selbst auch ein Breitenkreis (Kreis gleicher Breite ϕ). Bis auf den Äquator sind alle anderen Breitenkreise Kleinkreise auf der Kugel.

Die kürzeste Distanz $d(P_1, P_2)$ zwischen zwei Punkten $P_1(\lambda_1, \phi_1)$ und $P_2(\lambda_2, \phi_2)$ ist die Länge des Großkreisbogens zwischen diesen beiden Punkten und wird als **Großkreisdistanz** bezeichnet ([BC94]):

$$d(P_1, P_2) = R \cdot \arccos(\sin\phi_1 \cdot \sin\phi_2 + \cos\phi_1 \cdot \cos\phi_2 \cdot \cos(\lambda_1 - \lambda_2)),$$

mit R als dem Radius der Kugel.

Koordinaten auf der Ebene Lokationen auf der Ebene lassen sich sowohl über **Polarkoordinaten** als auch über **kartesische (rechtwinklige) Koordinaten** beschreiben. Ausgehend von einem beliebigen Punkt O auf der Ebene, dem Koordinatenursprung, kann ein beliebiger anderer Punkt P über seine Polarkoordinaten r und θ ausgedrückt werden. r ist der Abstand zum Ursprung und θ ist der Winkel bezüglich einer vorgegebenen Richtung (zumeist Nord). Ausgehend vom gleichen Ursprung und zwei rechtwinklig aufeinander stehenden Koordinatenachsen X (gerichtet nach Rechts, Ost) und Y (gerichtet nach Oben, Nord) lässt sich P auch über kartesische Koordinaten ausdrücken. Die klassischen kartesischen Koordinaten x und y beschreiben die jeweilige auf jeder Achse zurückzulegende Distanz, um vom Ursprung zu Punkt P zu gelangen. Punkt P lässt sich sowohl durch das Koordinaten-Tupel (r, θ), als auch über das Tupel (x, y) beschreiben. Polar- und kartesische Koordinaten lassen sich durch folgende Beziehungen ineinander überführen:

$$\begin{aligned} x &= r \cdot \sin\theta; \\ y &= r \cdot \cos\theta; \\ r &= \sqrt{x^2 + y^2}; \\ \theta &= \tan^{-1}\tfrac{y}{x}. \end{aligned}$$

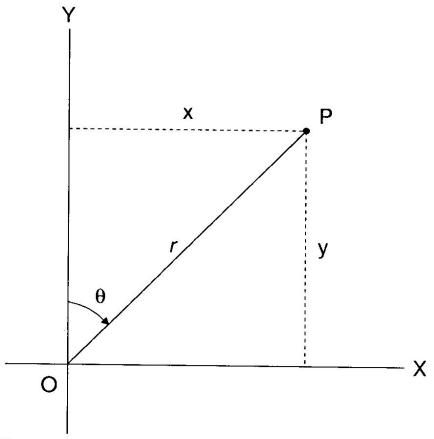


FIG. 4-2. Definition sketch for planar coordinates showing the relationship between polar and Cartesian types.

Abbildung 15: Beziehung zwischen kartesischen und Polarkoordinaten in der Ebene ([BC94]).

Der euklidische oder Phytagoräische Abstand $d(P_1, P_2)$ zwischen zwei Punkten $P_1(x_1, y_1)$ und $P_2(x_2, y_2)$ auf der Ebene basiert auf dem **Satz von Phytagoras** und ist definiert durch

$$d(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

Datum Nach Bill (2010; [Bil10]) spezifiziert das Datum eines Koordinatensystems die Beziehungen dieses Koordinatensystems zur Erde. Es beinhaltet die gegebenen Parameterwerte für das Koordinatensystem und definiert so die möglichen Freiheitsgrade. So wird z.B. festgelegt, welche(s) Erdfigur / Ellipsoid verwendet wird und Koordinatenursprung, Achsrichtungen, Maßstab und Projektionsverfahren werden definiert.

Kartenprojektion

Die meisten Karten und GIS verwenden kartesische, zweidimensionale Koordinatensysteme, um räumliche Sachverhalte auf der Erdoberfläche abzubilden. Dafür müssen die Punktlokationen auf der gekrümmten Ellipsoidoberfläche in Koordinaten auf der Ebene transformiert werden. Mit der **Kartennetzentwurfslehre** befasst sich eine eigene Wissenschaftsdisziplin mit der Entwicklung solcher Transformationsverfahren.

Die Abbildung der Ellipsoidoberfläche auf die Ebene wird grundsätzlich erreicht, indem man verschiedene Referenzflächen an das Ellipsoid so anlegt, dass die Fläche entweder in einem Punkt (Ebene) oder entlang einer Linie (Zylinder, Kegel) das Ellipsoid berührt. Ausgehend von einem Projektionszentrum (zumeist das Ellipsoidzentrum) wird jeder Punkt auf der Ellipsoidoberfläche auf die Referenzfläche projiziert. In der nachfolgenden Abbildung ist dies schematisch für zwei Punkte dargestellt. Drei-dimensionale Referenzflächen (Zylinder, Kegel) werden nach der Projektion so "ausgerollt", dass sie eine Ebene bilden. In einigen speziellen Projektionsverfahren schneidet die Referenzfläche das Ellipsoid, anstatt nur tangential anzuliegen (z.B. *Behrmanns Schnittzylinderentwurf*). Dadurch lassen sich in einem größeren Gebiet mögliche Verzerrungen minimieren. Jedes Projektionsverfahren besitzt eine mathematische Abbildungsvorschrift $(x', y') = f_{\text{Projektion}}(\lambda, \phi)$, welche es erlaubt, sowohl einen Punkt auf dem Ellipsoid in die Ebene abzubilden mit $P(\lambda, \phi) \rightarrow P'(x', y')$, als auch einen Punkt in projizierten Koordinaten wieder in seine sphärischen Koordinaten zu überführen mit $P' \rightarrow P$ mit $(\lambda, \phi) = f_{\text{inverse Projektion}}(x', y')$.

Referenzflächen Die verschiedenen Projektionsverfahren unterscheiden sich in der Art und Lage der verwendeten Referenzflächen. Zumeist kommen dabei Ebenen, Zylindermantelflächen oder Kegelmantelflächen zur Anwendung, da diese sich leicht mathematisch beschreiben lassen. Die **stereografische Projektion** (auch als **azimutale Abbildung** bezeichnet) verwendet eine Ebene,

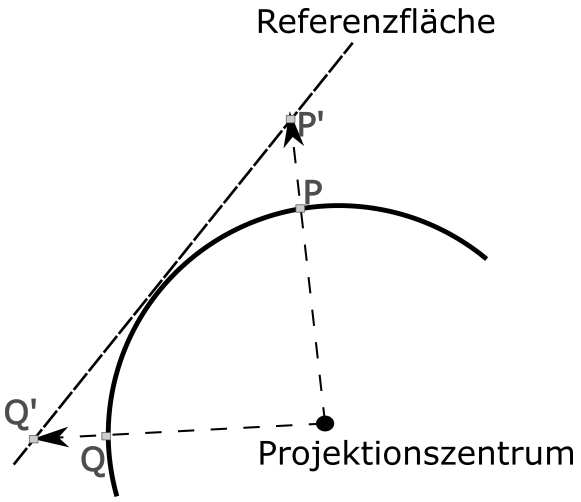


Abbildung 16: Projektion zweier Punkt P und Q von der Ellipsoidoberfläche auf eine Referenzfläche. Dadurch entstehen die projezierten Punkte P' und Q' auf der Ebene.

welche an die Erdfigur angelegt wird. Sie ist besonders für kleinräumigere Gebiete geeignet und wird in der Anwendung zumeist für Abbildungen der Polregionen genutzt, z.B. die *Universal Polar Stereographic (UPS)* - Projektion. Ausgehend vom Pol bilden hier die Meridiane konzentrische Geraden und die Breitenkreise nicht-äquidistante konzentrische Kreise. Bei einer **zylindrischen Projektion** ist die Referenzfläche eine an die Erdfigur angelegte Zylindermantelfläche. Dadurch lassen sich zum Beispiel die Meridiane als äquidistante Linien und die Breitenkreise als nicht-äquidistante Linie abbilden, welche sich im rechten Winkel kreuzen. Diese Abbildung ist auch für großräumige oder (fast) globale Abbildungen geeignet. Ein Beispiel hierfür ist die *Mercator*-Projektion. Eine Kegelmantelfläche dient bei der **konischen oder Kegel-Projektion** als Referenzfläche. Die Abbildung der Meridiane ergeben sich als Schnitte der Meridianebenen mit der Kegelfläche und sind Geraden. Die Breitenkreise werden als Kreise/Kreisbögen um das Kegelzentrum abgebildet. Ein Beispiel hierfür ist die *Lambert-Gaußsche winkeltreue Kegelprojektion*. Bezüglich der Lage der

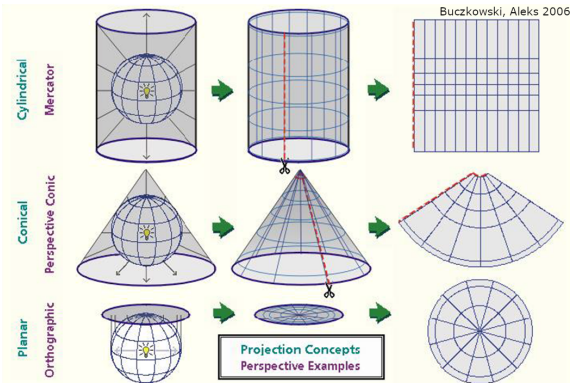


Abbildung 17: Verschiedene Projektionsansätze und die verwendeten Referenzflächen.

Referenzflächen wird zwischen **normalen**, **transversalen** und **schief-achsigen** Projektionen unterschieden. Bei einer normalen Abbildung wird die Referenzfläche entlang der Rotationsachse des Ellipsoids angelegt. Eine Ebene liegt dabei so, dass ihr Normalenvektor parallel zur Rotationsachse liegt. Zylinder- und Kegelmantelflächen liegen ebenfalls parallel zur Rotationsachse (siehe nachfolgende Abbildung). Liegen der Ebenen-Normalenvektor oder die Kegel-/Zylinderflächen senkrecht zur Rotationsachse, wird die Projektion als transversal bezeichnet.

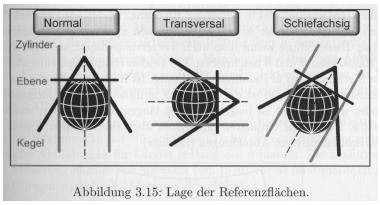


Abbildung 18: Lage der Referenzflächen ([Bil10]).

Abbildungseigenschaften und Verzerrung Koordinaten auf dem Ellipsoid lassen sich nie ganz verzerrungsfrei auf die Ebene abbilden. Diese Verzerrungen lassen sich über die so genannte **Tissotsche Indikatrix** beschreiben. Grafisch kann diese als ein Kreis mit festem Radius auf dem Ellipsiod repräsentiert werden, welcher durch die Abbildung auf die Ebene deformiert oder verzerrt wird. Die verschiedenen Projektionsverfahren lassen sich anhand ihrer Verzerrungseigenschaften in **winkeltreue (konforme)**, **längentreue** und **flächentreue** Verfahren klassifizieren.

Winkeltreue Projektionen (z. B. *Mercator*-Projektion, *Universal Polar Stereographic*-Projektion, *Lambert-Gaußsche winkeltreue Kegelprojektion*, *Gauß-Krüger*-Projektion, *Universal-Transversal-Mercator*-Projektion) erhalten die Winkelbeziehungen zwischen verschiedenen Objekten. Typischerweise treffen die Abbildungen der Meridiane und Breitenkreise im rechten Winkel aufeinander. Die Tissotsche Indikatrix bleibt zwar überall auf der Karte kreisförmig, allerdings verändert sich die Größe der Kreise abhängig von ihrer Position. Eine flächentreue Projektion (z.B. *Alberts-flächentreue*-Projektion, *Behrmanns Schnittzylinderentwurf*) erhält die Flächenrelationen der abgebildeten Objekte auf Kosten der Winkeltreue. Hier behalten die initialen Kreise der Tissotsche Indikatrix zwar ihre Fläche, werden jedoch zu Ellipsen verzerrt. Es ist nicht möglich, sowohl flächentreu, als auch winkeltreu abzubilden. Längentreue Abbildungen (z.B. *Quadratische Plattkarte*, *Azimutal-längentreue*-Projektion) sind weder flächen- noch winkeltreu, erhalten aber die Längenbeziehungen in bestimmte Richtungen. Die Tissotsche Indikatrix wird immer als Ellipse abgebildet. Das Verhältnis der großen zur kleinen Halbachse der Verzerrungsellipse ist in der längentreuen Richtung 1. Eine längentreue Abbildungen in alle Richtungen ist jedoch nicht möglich. Verläuft die längentreue Richtung entlang einer Ordinate (z. B. entlang der Längen - oder Breitenkreise), wird diese Abbildung als **ordinatentreu** bezeichnet ([Bil10]). Bestimmte längentreue Abbildungen werden auch als **vermittelnde** Abbildungen bezeichnet ([BC94]), wenn sie einen guten Kompromiss zwischen Winkel- und Flächenverzerrung bieten (z.B. *Robinson*-Projektion). Auf dieser Webseite zum Vergleich verschiedener Kartenprojektionen können Sie sich verschiedene existierende Projektionen mit ihren Eigenschaften anschauen und vergleichen.

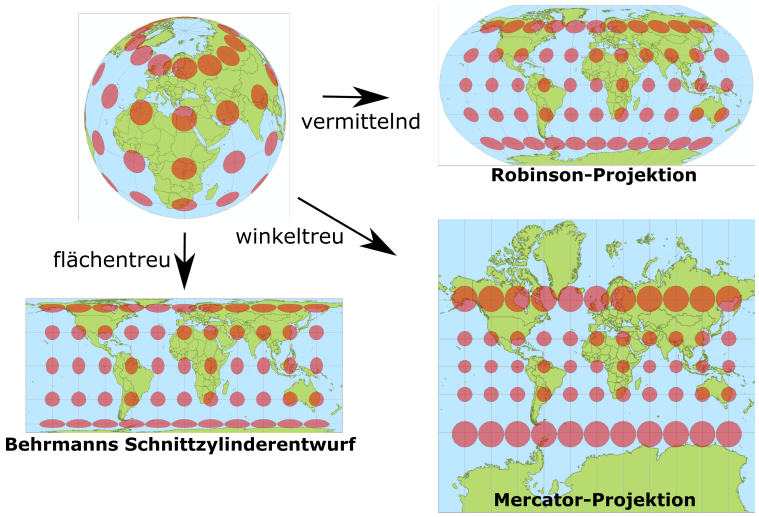


Abbildung 19: Tissotsche Indikatrix für eine flächentreue, eine winkeltreue und eine vermittelnde Projektion (https://de.wikipedia.org/wiki/Tissotsche_Indikatrix).

3.2 Koordinatentransformation und Georeferenzierung

Typische Eingangsdaten für ein GIS Projekt sind zum Beispiel

1. eine digitale geologische Karte, welche als Liste von Pixelkoordinaten mit Attributen vorliegt, mit bekannter Kartenprojektion und gegebenen Passpunkten
2. eine Tabelle mit geochemischen Messungen inklusive karthesiche Messpunktpositionen in Ost- (Easting) und Nord-Richtung (Northing) in einer anderen bekannten Projektion und
3. ein Satellitenbild in Rasterformat ohne Projektionsinformationen.

Diese drei Datensätze sollen im Zuge des GIS-Projektes aus ihrem initialen Raumbezug / Koordinatensystem in einen gemeinsamen Raumbezug überführt werden. Dieser ist innerhalb eines GIS typischerweise ein ebenes, katharisches Koordinatensystem (Projektkoordinaten - *working projection*; [BC94]). Dieses wird bei der Definition des GIS-Projektes gemäß der Aufgabenstellungen und Ziele definiert und korrespondiert meist nicht mit den Koordinatensystemen der Eingabedaten. Für jeden Datensatz müssen daher für diese Geocodierung in das Projektkoordinatensystem eine Reihe von **Koordinatenoperationen** durchgeführt werden. Dieser Workflow der Übertragung von Daten mit externem Raumbezug in den Raumbezug des GIS-Projektes wird in einem GIS häufig als **Georeferenzierung** bezeichnet und umfasst damit sowohl die Georeferenzierung in engeren Sinne (Zuweisen eines Raumbezug) als auch Geocodierung als Umwandlung von Raumbezügen ineinander.

Bei der geologischen Karte liegen für jeden Pixel initial nur die lokalen Pixelkoordiaten (u, v) vor. Diese werden in einem ersten Schritt (1) mittels der bekannten Passpunkte in die kartesischen Koordinaten (x, y) der bekannten Kartenprojektion transformiert. In einem zweiten Schritt (2) werden diese erst in geografische Koordinaten (λ, ϕ) und darauf basierend in einem dritten Schritt (2) in die karthesichen Koordinaten des Projektkoordinatensystems (x', y') umgewandelt. Für die Geocodierung in die Projektkoordinaten des zweiten Datensatzes sind nur die Umwandlungsschritte 2 und 3 notwendig, da die Punktdaten initial bereits in einer bekannten Projektion vorliegen. Das Raster-Satellitenbild muss in ein neues Rasterobjekt überführt werden, welches im Projektkoordinatensystem vorliegt. Dafür müssen die initialen lokalen Pixelkoordinaten (u, v) erst mittels Passpunkten am Boden in Projektkoordinaten (x', y') transformiert werden. Durch die Transformation liegen die Koordinaten der Pixelzentren danach nicht mehr notwendigerweise auf einem kartesischen Raster vor. Die Pixelwerte müssen daher auf die Pixel des neu angelegten Rasterobjekts, z.B. durch Interpolation, übertragen werden. Dieser Vorgang wird nach Bonham-Carter (1994, [BC94]) auch als *warping* oder *rubber-sheeting* bezeichnet.

Koordinatenoperationen

Im oben beschriebenen Beispiel werden zwei Koordinatenoperationen verwendet, um Koordinaten von einem Koordinatensystem in ein anderes zu überführen. Es handelt sich einerseits um die so genannte **Koordinatenumwandlung** und andererseits um die **Koordinatentransformation** mittels Passpunkten ([Bil10]).

Koordinatenumwandlung (*coordinate conversion*) Bei der Koordinatenumwandlung handelt es sich um die Umrechnung von gegebenen Koordinaten in einem bekannten Koordinatensystem in Koordinaten basierend auf einem anderen Koordinatensystem mittels bekannter Formalismen der Form $(x', y') = f(x, y)$. Ein Beispiel ist hier die Überführung von ebenen Polarkoordinaten in ebene kartesische Koordinaten wie in Abschnitt Koordinatensysteme und Kartenprojektion gezeigt. Innerhalb eines GIS erfolgt die Umwandlung zumeist von Koordinaten in einer bekannten ebenen Projektion $(x', y') = f_{\text{Projektion 1}}(\lambda, \phi)$ in Koordinaten in einer einer anderen bekannten ebenen Projektion $(x'', y'') = f_{\text{Projektion 2}}(\lambda, \phi)$. Da nicht für jeder Projektionskombination explizite Umwandlungsformalismen vorliegen, werden die gegebenen ebenen Koordinaten in einem Zwischenschritt in geografische Koordinaten umgewandelt, welche dann wieder neu projiziert werden können:

$$(x', y') \rightarrow f_{\text{inverse Projektion 1}}(x', y') = (\lambda, \phi) \rightarrow f_{\text{Projektion 2}}(\lambda, \phi) = (x'', y'')$$

Ebenso lassen sich zum Beispiel geografische Koordinaten basierend auf einem Referenzellipsoid in geografische Koordinaten basierend auf einem anderen Ellipsoid umwandeln.

Koordinatentransformation (coordinate transformation) Koordinatentransformation wird immer dann verwendet, wenn kein expliziter Formalismus bekannt ist, um von einem Koordinatensystem in ein anderes umzuwandeln. Am häufigsten tritt dieser Fall auf, wenn die initialen Daten in lokalen Koordinaten ohne Anschluss an ein bekanntes Referenzsystem vorliegen und in ein bekanntes Referenzsystem übertragen werden sollen. Für Rasterbilder ist zwar ggf. eine Projektion bekannt, aber die Koordinaten für die Pixel liegen nur in lokalen, relativen Pixelkoordinaten vor. In diesen Fällen müssen die Parameter für die Umwandlungsvorschrift basierend auf Passpunkten geschätzt werden.

Passpunkte sind einzelne Tupel von Koordinatenpaaren $\{(x, y), (x', y')\}$, für die sowohl die Ausgangskoordinaten (x, y), als auch die Zielkoordinaten (x', y') bekannt sind (siehe folgende Abbildung zu Koordinaten-basierten Passpunkten).

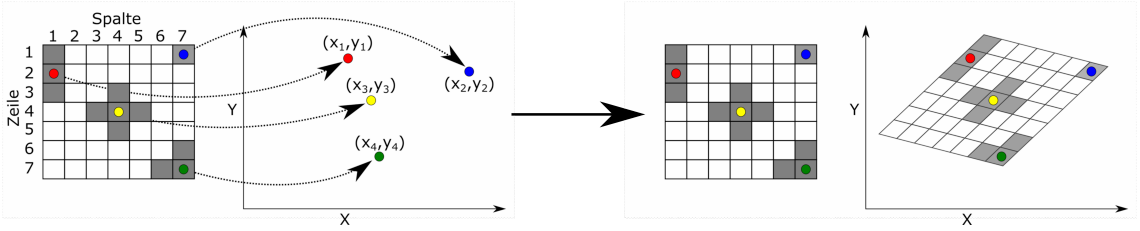


Abbildung 20: Bestimmung der Transformationsparameter über Koordinaten-basierende Passpunkte.

Alternativ kann man Passpunkte auch bestimmen, in dem Objekte im zu transformierenden Datensatz mit Objekten in einem bereits in Zielkoordinatensystem vorliegenden Datensatz assoziiert werden. Für die assoziierten Objekte wird angenommen, dass sie sich in der Realität an der gleich Position befinden sollen (siehe folgende Abbildung zu Objekt-basierten Passpunkten). Wenn zum Beispiel in beiden Datensätzen das gleiche, markante Gebäude zu identifizieren ist, so kann dieses für die Bestimmung eines Passpunktes verwendet werden. In diesem Fall muss sowohl das Ziel- als auch die Ausgangskoordinatensystem nicht notwendigerweise bekannt sein. Es ist also möglich, aus einem unbekannten Koordinatensystem in ein anderes unbekanntes Koordinatensystem zu transformieren, wenn sich solche Referenzobjekt für beide Koordinatensysteme finden.

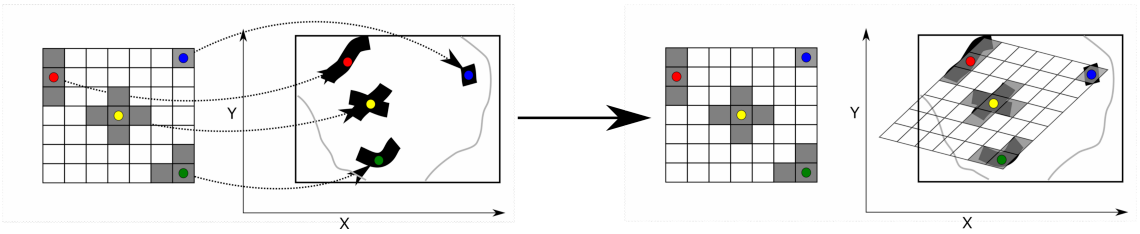


Abbildung 21: Bestimmung der Transformationsparameter über Objekt-basierende Passpunkte.

Für die Transformation von $(x, y) \rightarrow (x', y')$ werden im Allg. empirische polynomiale Funktionen niedrigen Grades verwendet:

Grad 1: $\begin{cases} x' = a_1 + b_1 \cdot x + c_1 \cdot y \\ y' = a_2 + b_2 \cdot x + c_2 \cdot y \end{cases}$

Grad 2 (quadratisch): $\begin{cases} x' = a_1 + b_1 \cdot x + c_1 \cdot y + d_1 \cdot x^2 + e_1 \cdot y^2 + f_1 \cdot x \cdot y \\ y' = a_2 + b_2 \cdot x + c_2 \cdot y + d_2 \cdot x^2 + e_2 \cdot y^2 + f_2 \cdot x \cdot y \end{cases}$

Grad 3 (kubisch): $\begin{cases} x' = a_1 + \ldots + g_1 \cdot x^3 + h_1 \cdot y^3 + i_1 \cdot x^2 \cdot y + j_1 \cdot x \cdot y^2 \\ y' = a_2 + \ldots + g_2 \cdot x^3 + h_2 \cdot y^3 + i_2 \cdot x^2 \cdot y + j_2 \cdot x \cdot y^2 \end{cases}$

Grad 4: $\begin{cases} x' = a_1 + \ldots + k_1 \cdot x^4 + l_1 \cdot y^4 + m_1 \cdot x^3 \cdot y + n_1 \cdot x \cdot y^3 + o_1 \cdot x^2 \cdot y^2 \\ y' = a_2 + \ldots + k_2 \cdot x^4 + l_2 \cdot y^4 + m_2 \cdot x^3 \cdot y + n_2 \cdot x \cdot y^3 + o_2 \cdot x^2 \cdot y^2 \end{cases}$

...

Die freien Parameter (Koeffizienten; *coefficients*) $\{a_1, b_1, c_1, \ldots, a_2, b_2, c_2, \ldots\}$ können über die gegebenen Passpunkte bestimmt werden, indem folgende Gleichungssysteme gelöst werden:

$$\begin{bmatrix} 1 & x_1 & y_1 & \ldots \\ 1 & x_2 & y_2 & \ldots \\ 1 & x_3 & y_3 & \ldots \\ & \ddots & & \ldots \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ \vdots \end{bmatrix} \text{ und } \begin{bmatrix} 1 & x_1 & y_1 & \ldots \\ 1 & x_2 & y_2 & \ldots \\ 1 & x_3 & y_3 & \ldots \\ & \ddots & & \ldots \end{bmatrix} \cdot \begin{bmatrix} a_2 \\ b_2 \\ c_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \\ \vdots \end{bmatrix}.$$

Daraus ergibt sich, dass immer mindestens halb so viele Passpunkte wie freie Parameter benötigt werden, um diese eindeutig zu bestimmen. Werden mehr Passpunkte verwendet, wird das entstehende Gleichungssystem über die Methode der kleinsten Quadrate"gelöst. Dies erlaubt es, zusätzlich einen Projektionsfehler zu bestimmen. Es empfiehlt sich demzufolge, immer mehr als die notwendige Mindestanzahl an Passpunkten zu verwenden, so dies denn möglich ist.

Für eine polynomiale Transformation 1. Grades sind aufgrund der **6 freien Parameter** also **mindestens 3 Passpunkt-Tupel** $\{(x_1, y_1), (x'_1, y'_1)\}$,$\{(x_2, y_2), (x'_2, y'_2)\}$, $\{(x_2, y_2), (x'_2, y'_2)\}$ notwendig. Transformationen 1. Grades werden auch als "**affine Transformationen**" (affin = "gradentreu") bezeichnet. Sie erlauben die Berücksichtigung von **Skalierung**, **Verschiebung** (Translation) des Koordinatenursprungs und **Drehung** (Rotation) um den Koordinatenursprung zwischen zwei Koordinatensystemen. Die zu lösenden Gleichungssysteme vereinfachen sich im Fall von nur 3 Passpunkten zu den zwei linearen Gleichungssystemen

$$\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} \text{ und } \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \cdot \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix} = \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \end{bmatrix}.$$

Die affinen Projektionsgleichungen lassen sich auch als Vektorform ausdrücken:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R \cdot S \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \vec{t},$$

mit der Skaliermatrix $S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$, der Rotationsmatrix $R = \begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix}$ und dem Verschiebungsvektor $\vec{t} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$. Die Koeffizienten $\{b_1, c_1, b_2, c_2, \}$ ergeben sich dann durch Matrixmultiplikation und auf Grund der diagonalen Struktur von S wie folgt:

$$\begin{bmatrix} b_1 & c_1 \\ b_2 & c_2 \end{bmatrix} = R \cdot S = \begin{bmatrix} b_1 = s_x \cdot r_1 & c_1 = s_y \cdot r_2 \\ b_2 = s_x \cdot r_3 & c_2 = s_y \cdot r_4 \end{bmatrix}.$$

Da Matrixmultiplikation NICHT kommutativ ist, ist es entscheidend, dass S von rechts an R multipliziert wird. Die Reihenfolge der Faktoren muss immer beachtet werden.

Vor allem im Fall von unbekannten Koordinatensystemem werden häufig Transformationen höheren Grades benötigt, da diese zusätzliche zu Skalierung, Verschiebung und Rotation zusätzlich **Verzerrungs- und Scherungseffekte** berücksichtigen können. Durch die deutlich höhere Anzahl an freien Parametern erhöht sich dabei allerdings die Anzahl der mindestens notwendigen Passpunkte:

Grad 2: 12 freie Parameter ⇒ min. 6 Passpunkte

Grad 3: 20 freie Parameter ⇒ min. 10 Passpunkte

Grad 4: 30 freie Parameter ⇒ min. 15 Passpunkte

Georeferenzierung von Vektordaten

Der folgende Workflow illustriert die Georeferenzierung von drei Vektordatensätzen (A, B, C) nach Bonham-Carter (1994, [BC94]). Die Daten liegen jeweils als Koordinatenlisten mit Attributen vor. Zu beachten ist hier, dass Bonham-Carter nicht zwischen Koordinatentransformation und Koordinatenumwandlung unterscheidet. Beide Operationen werden als *convert* (Konvertierung, Umwandlung) bezeichnet. Bei den für Datensatz A bekannten *table coordinates* (u, v) handelt es sich um lokale, relative Koordinaten eines Digitalisiertisches, welche als analog zu lokalen Pixelkoordinaten digitaler Rasterbilder betrachtet werden können.

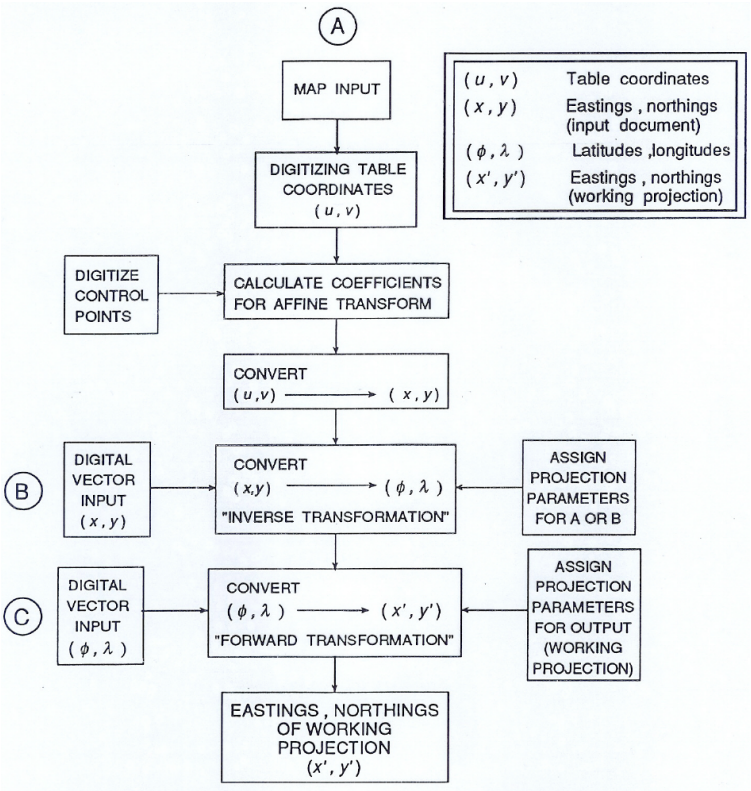


FIG. 4-10. Steps in converting vector data to the planar Cartesian coordinates of a working GIS projection. **Source A** consists of Cartesian coordinates from a digitizing table. **Source B** consists of eastings and northings in the projection of the input document. **Source C** is similar to **B**, except that the coordinates are already in geographic coordinates.

Abbildung 22: Bonham-Carter (1994, [BC94])

Georeferenzierung von Rasterdaten

Der folgende Workflow illustriert die Georeferenzierung eines Rasterdatensatzes (*input grid*) nach Bonham-Carter (1994, [BC94]). Die Pixelzentren werden über die Zeilennummer $R = 1 \leq r \leq n_r$ und die Spaltennummer $C = 1 \leq c \leq n_r$ identifiziert, der Pixelattributwert ist Z. Die räumliche Lage jedes Pixel im untransformierten Raster ergibt sich relativ ausgehend von einer bekannten Ursprungskoordinate $O = (x_0, y_0)$ und einer bekannten Pixelkantenlänge Δ. In einfachsten Fall entspricht der Ursprung dem Zentrum des ersten Pixels ($r = 0, c = 0$). Die Lage eines beliebigen Pixelzentrums kann dann über $(x_r, y_r) = (x_0 + c \cdot \Delta, y_0 + r \cdot \Delta)$ bestimmt werden. Nach der Transformation der Koordinaten aller Pixelzentren gilt diese Beziehung nicht mehr, da die projizierten Pixelzentren kein regelmäßiges kartesisches Raster mehr bilden. Um aber weiterhin auf einem kartesischen Raster arbeiten zu können, werden die Attributwerte an den projizierten Pixeln auf die Pixel eines neuen Rasters (*output grid*) übertragen, dessen Ursprung im projizierten Koordinatensystem definiert ist und dessen Pixel rechtwinklig und äquidistant zu den Koordinatenachsen dieses Koordinatensystems angeordnet sind.

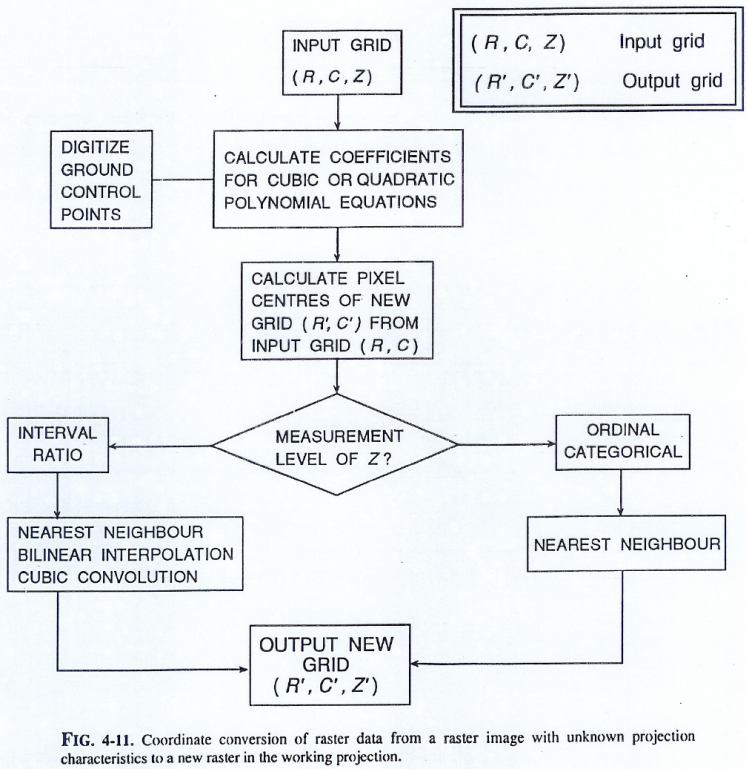


FIG. 4-11. Coordinate conversion of raster data from a raster image with unknown projection characteristics to a new raster in the working projection.

Abbildung 23: Bonham-Carter (1994, [BC94])

4 Modellierung räumlicher Daten

Der Prozess der **Datenmodellierung** umfasst die abstrahierende Beschreibung realer Sachverhalte, um diese mittels eines Computers verarbeiten und speichern zu können. Dafür wird ein zu beschreibender komplexer Sachverhalt durch ein vereinfachtes oder verallgemeinertes Modell abgebildet. Diese Vereinfachung / Verallgemeinerung erfolgt dabei aus fach- oder problemspezifischer Sicht, d.h. es werden vorzugsweise die für ein Fachgebiet oder eine Problemstellung relevanten Eigenschaften eines Sachverhaltes betrachtet ([Bil10]). Diese Eigenschaften werden dann auf eine Art und Weise organisiert, dass sie sich in einem konsistenten Datensatz überführen lassen, der einerseits digital verarbeitet werden kann und aus dem andererseits weitergehende Informationen über den betrachteten Sachverhalt abgeleitet werden können ([BC94]). Für das in Abschnitt *Einführung* vorgestellte Problem des digitalen Höhenmodells eines Untersuchungsgebietes bedeutet dies, dass die komplexe "wahre" Höheninformation über diesem Gebiet einerseits in einer realisierbaren Weise an diskreten Punkten gemessen und als Zahlenwerte gespeichert werden muss. Die ursprünglich kontinuierliche Höheninformationen liegt nach der Messung als endliche Menge von Messlokationen und Höhenwerten vor. Dies ist zwar eine Abstraktion der wahren Höheninformation im Untersuchungsgebiet, kann aber als repräsentativ für diese angesehen werden.

Ein **Datenmodell** ist das Ergebnis einer konzeptionellen Datenmodellierung und beschreibt die tatsächliche Organisation der Daten bezügliches eines vorgegebenen Schemas. Wenn für das Beispiel des DGM die Messlokationen ausgehend von einem bekannten Punkt in einem äquidistanten Gitter vorliegen, kann man für diesen Datensatz das so genannte Rastermodell verwenden. Die Höheninformation liegt konzeptionell als aus einem äquidistanten Raster gleichförmiger Flächenelemente (Pixel) vor, jeder Höhenwert gilt für ein Pixel. Die Messlokationen werden mit den Pixelzentren assoziiert und lassen sich implizit aus Ausgangskoordinate und Gitterschrittweise herleiten.

Eine **Datenstruktur** repräsentiert ein solches Datenmodell. Es handelt sich um die implementatorisch-technische Umsetzung dieses Modells für eine tatsächliche Anwendung. Das Pixelraster des DGM könnte zum Beispiel über eine Matrix repräsentiert werden. Jeder Matrixeintrag entspricht einem Höhenwert, dessen Position in der Matrix (Zeilen- und Spalten) mit der Position des zugehörigen Pixels im Raster korrespondiert. Um diese Datenstruktur speichern zu können, muss ein für diese Struktur angemessenes **Datenformat** (*file format*; [BC94]) verwendet werden. Eine Möglichkeit zur Speicherung des DEM-Rasters wäre zum Beispiel hier die Speicherung aller Werte einer Matrixzeile als Liste durch Semikolon getrennter Zahlenwerte pro Dateizeile und der Rasterinformationen (Ausgangskoordinaten und Pixel) als Header-Zeilen. Die Zieldatei hätte also ein bis zwei Zeilen mit Zusatzinformationen und so viele Datenzeilen, wie die Datenmatrix Zeilen ausweist, mit jeweils so vielen Werten pro Zeile, wie die Datenmatrix Spalten aufweist.

4.1 Räumliche Objekte (*spatial objects*)

Wie bereits in Abschnitt *Geoinformationssysteme* eingeführt, werden Daten in einem GIS in sogenannten Geoobjekten gruppiert. Geoobjekte stellen Modelle realer Sachverhalte dar und sind zusammenhängende, räumlich abgeschlossene Einheiten mit ähnlichen thematischen Eigenschaften oder gleicher oder ähnlicher Bedeutung. Anhand ihrer Dimension lassen Sie sich in

Punktobjekte Dimension 0, 0D,

Linienobjekte Dimension 1, 1D,

Flächenobjekte Dimension 2, 2D und

Volumenobjekte Dimension 3, 3D

unterteilen. Flächenobjekte mit eindeutig zugeordnetem Parameterwert (z. B. Höhe) werde manchmal auch als 2.5-dimensional bezeichnet ([BC94]). Eindeutig zugeordnet bedeutet in diesem Fall, dass für jede 2D-Koordinate exakt nur ein Wert für einen Parameter auftritt. Es darf keine weitere Lokation mit der gleichen 2D-Koordinate und einem anderem Parameterwert auftreten. Dies ist nur für allgemeine oder *echte* 3D-Objekte zulässig.

Natürliche und künstliche räumliche Objekte

In der Realität können messbare Sachverhalte, Eigenschaften oder Parameter entweder **kontinuierlich** (z. B. Temperatur oder Schwerkraft) oder **diskret** (z.B. Lithologie) auftreten. Bei der an gegebenen Punkten gemessenen Temperatur handelt es sich um eine **kontinuierliche Feldvariable**, wohingegen es sich bei der Lithologie um eine **diskrete Klassenvariable** handelt. Feldvariablen können theoretisch unendlich viele unterschiedliche Werte annehmen, Klassenvariablen nur Werte aus einer gegebenen endliche Wertemenge. Parameterkontraste äußern sich bei Feldvariablen durch lokalisierte hohe Gradienten bei weiterhin stetigem Feldverlauf, wohingegen diskrete Variablen lokalisierte Sprünge zwischen den Werten aufweisen können.

Räumliche Einheiten, welche durch scharfe Parameterkontraste von ihrer Umgebung abgrenzbar sind, können innerhalb eines Datenmodells als **natürliche unregelmäßig-geformte räumliche Objekte** behandelt werden *(natural irregular shaped spatial objects*; [BC94]). Regionen mit kontinuierlichen Parametern müssen dafür in diskrete räumliche Objekte unterteilt werden, in denen die Variable entweder als konstant angenommen wird, oder formalisierbar repräsentiert ist. Diese Unterteilungsobjekte können unregelmäßig (z.B. Polygone) oder gleichförmig (z. B. Pixel, Gitterzellen) geformt sein. Jedes dieser Unterteilungsobjekt ist für sich selbst ein räumliches Objekt , die Menge aller verbunden Unterteilungsobjekte (topologische Vermaschung oder Raster) ist aber ebenfalls ein räumliches Objekt.

Natürliche räumliche Objekte korrespondieren mit realen diskreten natürlich-auftretenden räumlichen Einheiten, z. B. ein Fluss oder ein Erzkörper. **Künstliche räumliche Objekte** (*imposed spatial object*) sind einerseits künstlich abgeleitete Einheiten, wie z.B. Pixel, Dreiecke, Linien gleicher Eigenschaftswerte (Isolinien) und andererseits Objekte, welche vom Menschen "geschaffen" wurden. Physisch-auftretende künstliche räumliche Einheiten sind zum Beispiel Straßen oder Gebäude. Nicht-physische Regionen wurden definiert, z.B. Verwaltungsbezirkte oder Flurstückgrenzen. Sie stehen nicht unbedingt in Beziehung zu natürlich-auftretenden oder physisch-auftretenden Objekten ([BC94]).

Auflösungs-begrenzte (*sampling-limited*) und Definitions-begrenzte (*definition-limited*) natürliche räumliche Objekte

Objekte, deren Form und Ausdehnung allein abhängig von der Position und Anzahl der Messlokationen (*sampling*) sind, werden als Auflösungs-begrenzte (*sampling-limite*d) Objekte bezeichnet. Die Form einer Linie, welche eine Küstenlinie repräsentieren soll, ist zum Beispiel davon abhängig, wo und in welchen Abständen die Position dieser Küstenlinie gemessen wurde. Zusätzliche Attribute beeinflussen die Form und Ausdehnung nicht. Im Gegensatz dazu ist die Form und Ausdehnung von Definitions-begrenzten (*definition-limited*) Objekten zusätzlich von vom Anwender spezifizierten Grenzwerten für Attribute abhängig. So ist die Form einer "Blei-kontaminierten" Regionen sowohl von der Position der Messlokationen, als auch von den gemessenen Bleiwerten und den für eine Kontamination geltenden Grenzwerten abhängig.

Unregelmäßige und regelmäßige künstliche räumliche Objekte

Künstliche Objekte wie Verwaltungsbezirke, Grundstücksgrenzen oder Straßen sind grundsätzlich unregelmäßig geform. Künstlich abgeleitete Unterteilungsobjekte können ebenfalls unregelmäßig sein, wenn sie auf einer nicht-gleichförmigen Zerlegung basierend, z.B. allgemeine Dreiecke (TIN - *triangulated irregular network*), Polygone oder Voronoi-Zellen. Unregelmäßig verteilte Messpunkte oder Schnitte an beliebigen, nicht-ebenen Flächen gehören ebenfalls zu dieser Klasse von künstlichen Objekten.

Jede regelmäßige Unterteilung des Raums oder der Fläche erzeugt eine Menge gleichartiger regelmäßiger Objekte, z.B. quadratische Pixel in Rasterbildern, oder strukturierte Raster aus Hexaedern oder gleichseitigen Dreiecken. Zu dieser Klasse von Objekten gehören aber auch Messpunkte auf einem regelmäßigen Messgitter oder Schnitte an einer ebenen Fläche ([BC94]).

Table 2-1. Some geological examples of spatial objects, organized by type and by spatial dimension. Objects that are naturally-occurring can be organized according to whether they are sampling limited or definition limited. Objects that are imposed are either regular (like the pixels in a raster image), or irregular, like the polygons in a Voronoi diagram. Naturally occurring spatial objects of 1 or more dimensions are virtually always irregular in shape (exception: columnar basalt).

TYPE		SPATIAL DIMENSION 0-D POINT	1-D LINE	2-D AREA	2.5-D SURFACE*	3-D VOLUME
NATURALLY OCCURRING	SAMPLING LIMITED	lineament intersection	inferred contact	flood zone	top of coal seam in subsurface	salt dome
	DEFINITION LIMITED	seismic epicentre	contour line	geochemical anomaly	thermocline	ore body
IMPOSED	IRREGULAR	soil sample locations	drill hole in vertical cross-section	mining claim	non-planar cross-section	3-D excavation
	REGULAR	sample locations on grid	flight line traverse	grid cell	planar cross-section	voxel

*Single-valued surfaces, such that any (x,y) location has only a single value of z. A folded surface would therefore not qualify, because multiple values of z occur at given (x,y) locations.

Abbildung 24: Beispiele für räumliche Objekte nach Bonham-Carter (1994, [BC94]).

4.2 Datenmodelle

4.2.1 Rastermodell

Im Rastermodell wird ein Gebiet in ein Gitter aus regelmäßigen Flächen- oder Volumenelementen, so genannten Zellen, zerlegt. Sind diese Zellen quadratisch oder zumindest rechteckig, werden sie auch als Pixel (2D) oder Voxel (3D) bezeichnet. Andere regelmäßige Formen, wie Hexaeder oder gleichseitige Dreiecke/Tetraeder sind möglich, aber eher unüblich. Die Position jedes Pixels wird über die Zeilen- und Spaltennummer im Gitter adressiert. Die Kantenlänge eines Pixels und damit die überdeckte Fläche wird bei der Rastererstellung definiert und gilt für alle Pixel eines Rasterdatensatzes. Durch die Kantenlänge ergibt sich die Auflösung des Rasters. Ausgehend von quadratischen Pixeln bedeutet eine **Auflösung** von 100 m, dass die Überdeckung eins quadratischen Gebietes von 100 km^2 ein Raster von 1000 Zeilen und 1000 Spalten und damit einer Million Pixeln benötigt. Bei einer Auflösung von 10 m benötigt ein Raster mit der gleichen Überdeckung schon jeweils 10 000 Zeilen und Spalten mit insgesamt 100 Millionen Pixeln. Üblicherweise werden Rasterdaten über Matrizen ihrer Attributwerte und ggf. einigen Metainformationen repräsentiert. Dies ist allerdings sehr speicheraufwändig, da für jeden Pixel mindestens ein Matrix-Eintrag vorgehalten werden muss. Um den Speicheraufwand zu minimieren, kommen Kompressionsverfahren, wie zum Beispiel *Lauflängen-Codierung* (siehe *Datenstrukturen für Rasterdaten*) zum Einsatz.

Die räumliche Lage jedes Pixels liegt nur implizit vor. Ausgehend von einer bekannten räumlichen Ursprungskoordinate, welche zumeist an eine Ecke oder das Zentrums eines Pixels gekoppelt ist und der bekannten Auflösung lässt sich die räumliche Lage jedes Pixels herleiten und muss nicht dauerhaft gespeichert werden. Durch die Anordnung der Zellen in einem Gitter ist es sehr leicht möglich, Nachbarschaftsbeziehungen zwischen Pixeln abzuleiten. Als echte Nachbarn eines Pixels p gelten alle anderen Pixel, welche an eine Kante von p angrenzen, bzw. in ihrer Zeilennummer oder Spaltennummer um ± 1 von p abweichen. Echte Nachbarn von $p = (u, v)$ weisen somit die Gitterkoordinaten $(u \pm 1, v)$ oder $(u, v \pm 1)$ auf. Die folgende Abbildung illustriert dies nochmals schematisch.

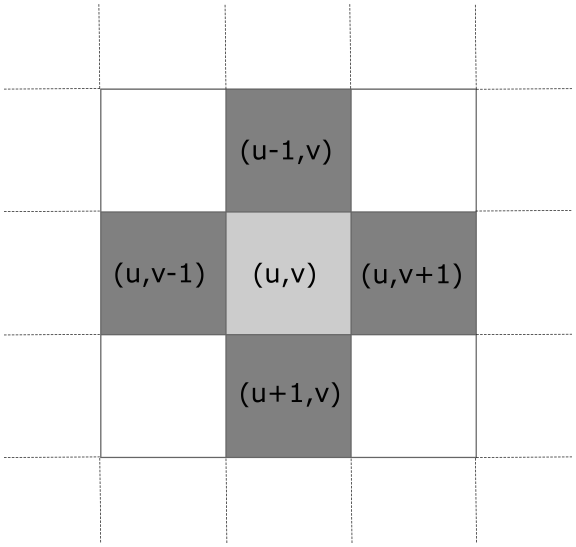


Abbildung 25: Schematische Darstellung der echten Nachbarn (dunkel-grau) eines Pixels (u, v)

Pro Pixel wird typischerweise nur ein Attributwert gespeichert. Ein Objekt im Rastermodell wird deshalb in so genannte Bänder unterteilt. Jedes Band wird einem Attribut zugeordnet und speichert die Attributwerte für jedes Pixel in einer Matrix. So besitzen klassische Farbrasterbilder drei Bänder für die drei Farbkanäle Rot, Grün und Blau. Für jedes Band eines Rasterbildes sind Auflösung und räumlicher Ursprung identisch.

Objekte im Rastermodell sind immer flächenhafte Objekte aus zusammenhängenden Pixeln, die diesem Objekt zugeordnet sind, zum Beispiel über einen gemeinsamen Attributwert. Ein Objekt mit dem Attribut `ID=27` wird so zum Beispiel über alle Pixel definiert, welche für das Attribut `ID` den Wert `ID=27` aufweisen (siehe nachfolgende Abbildung). Im Rastermodell gibt es keine explizite Unterscheidung zwischen Punkten, Linien und Flächen. Punktförmige Sachverhalte lassen sich zwar durch einzelne Pixel mit einem von den Nachbarpixeln abweichenden Attribut identifizieren und linienförmige Sachverhalte durch Ketten zusammen hängender Pixel mit gleichem Attributwert. Dies ist jedoch nur eine Approximation, da diese Objekte im Rastermodell, bedingt durch die von den Pixeln überdeckte Fläche, dennoch immer flächenhaft vorliegen. Die Identifizierung einzelner unbekannter Objekte aus einem Rasterdatensatz ist möglich, aber zum Teil sehr aufwändig.

Räumliche Berechnung, z.B. Fläche oder Umfang von Objekten, basieren im Rastermodell allein auf dem **Abzählen** von Pixeln und sind so sehr effizient. Die Fläche eines Objektes mit dem Attribut `ID=27` ergibt sich durch das einfache Abzählen aller Pixel mit diesem Attributwert und der Multiplikation dieser Anzahl mit der durch die Auflösung bekannten überdeckten Pixelfläche:

$$\text{Fläche eines Objektes} = \text{Anzahl der Pixel} \cdot \text{Fläche eines Pixels}.$$

Der Umfang dieses Objektes wird ermittelt, im dem für jeden zum Objekt zugeordneten Pixel gezählt wird, wie viele Nachbarpixel mit einem abweichenden Attributwert er besitzt. Die Anzahl aller dieser Nachbarpixel wird dann mit der bekannten Kantenlänge eines Pixels multipliziert:

$$\text{Umfang eines Objektes} = \Big(\sum_{\text{Anzahl der Pixel}} \text{Anzahl der Nachbarpixel mit anderem Attributwert} \Big) \cdot \text{Kantenlänge eines Pixels}.$$

Das Rastermodell kommt in erster Linie für digitale Bilddaten zur Anwendung. Das sind zum Beispiel klassische RGB-Luft- und Satellitenbilder, sowie Multi-Spektral-Bilder aus dem Remote-Sensing. Auch digital erstellte oder gescannte Karten werden zumeist im Rastermodell repräsentiert. Zusätzlich lassen sich kontinuierliche Sachverhalte sehr gut über Rasterdaten abbilden. Für die Bearbeitung und Analyse solcher Bilddaten stehen sehr leistungsfähige Methoden zur Verfügung. Dreidimensionale Rasterdaten werden vor allem im medizinischen oder technischen Bereich (Materialwissenschaften) verwendet. Die Eigenschaften von Daten im Rastermodell lassen sich wie folgt zusammenfassen ([Bil10]):

- Grundelement: gleichförmige Flächen- oder Volumenzelllen wie Pixel oder Voxel
- Ausschließlich Flächenhafte Betrachtungsweise von Objekten
- Einzelobjekte lassen sich nur über Attribute abgrenzen
- Position und Nachbarschaftsbeziehungen von Pixeln sind über die Gitteranordnung definiert
- Datenerfassung ist einfach und effizient
- Sehr speicheraufwändig

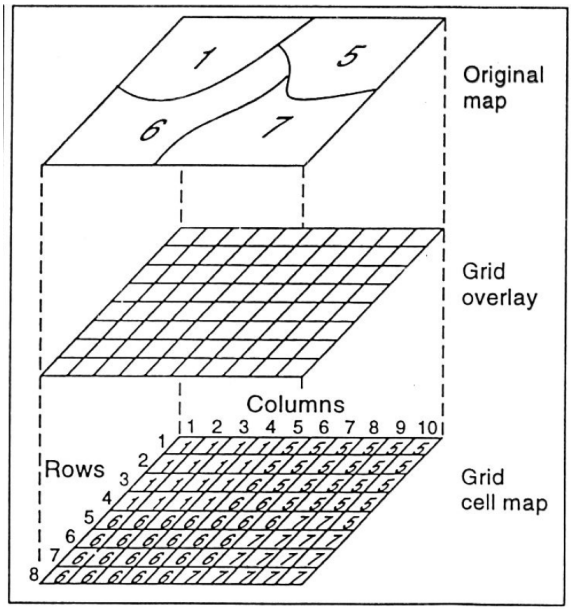


Abbildung 26: Objekte repräsentiert im Rastermodell ([BC94]). Jedem Pixel wird eines der initialen Objekte zugeordnet

4.2.2 Vektormodell

Im Vektormodell liegt das Hauptaugenmerk in der möglichst exakten Repräsentationen der räumlichen Lage und Ausdehnung von unregelmäßigen räumlichen Objekten. Das verwendete Basiselement ist ein so genannter **Vertex** $\vec{v}$, der über das Tupel seiner Punktkoordinaten, im katharischen Fall $\vec{v} = (x, y)$, definiert wird. Mathematisch kann dies als Vektor aufgefasst werden, daher rührt auch die Bezeichnung **Vektor**modell. Über Vertices lassen sich punktförmige Objekt exakt beschreiben. Beliebige linienförmige Objekte werden als Sequenz von Liniensegmenten (Linienzug) $\{s_1, s_2, \ldots\}$ dargestellt. Jedes Liniensegment verknüpft zwei Punktkoordinaten mit $s = \{(x_{Anfang}, y_{Anfang}), (x_{Ende}, y_{Ende})\}$, wobei typischer Weise gilt, dass der Endvertex eines Liniensegmentes mit dem Startvertex des darauffolgenden Liniensegments übereinstimmt mit $s_i(x_{Ende}, y_{Ende}) = s_{i+1}(x_{Anfang}, y_{Anfang})$. Diese Kette von Liniensegmenten wird auch als *String* bezeichnet ([BC94]). Im Allgemeinen werden für einen Linienzug die Koordinaten nicht doppelt, sondern als einfache Liste der Anfangsknoten der Segmente gespeichert. Flächenobjekte werden im Vektormodell als so genannte Polygone (beliebige Vielecke) repräsentiert. Ein Polygon wird dabei über den geschlossenen Linienzug (*Loop*; [BC94]) seiner Grenze beschrieben. Eine solche Loop unterscheidet sich von einem einfachen Linienzug dahingehend, dass der Endvertex des letzten Segmentes mit dem Anfangsvertex des ersten Segmentes übereinstimmt mit $s_n(x_{Ende}, y_{Ende}) = s_1(x_{Anfang}, y_{Anfang})$, der Linienzug ist somit geschlossen. Analog dazu werden dreidimensionale Objekte (Raumkörper, Polyeder) über ihre Grenzflächen definiert.

Die Verwendung von exakten Koordinaten ermöglicht mathematisch exakte räumliche Berechnungen. Die Länge eines Linienzuges wird dabei über die Summe der Länge der Einzelsegmente berechnet mit

$$\text{Länge} = \sum_{i=1}^{n} d(s_i),$$

wobei im kartesischen Fall $d(s_i)$ über

$$d(s_i) = \sqrt{(x_{Ende} - x_{Anfang})^2 + (y_{Ende} - y_{Anfang})^2}$$

bestimmt wird. Auf die gleiche Weise lässt sich der Umfang eines Flächenobjektes über die Länge seiner Grenzlinie bestimmen. Die Fläche eines Polygons, welches über eine geordnete Liste der Punkte seiner Grenzlinie beschrieben ist, kann wie folgt bestimmt werden. Sei $P_i = (x_i, y_i)$, mit $i = 1, \ldots, n$ und $P_1 = P_{n+1}$ ein solcher Grenzvertex und $P = (x, y)$ ein beliebiger Punkt im Inneren des Polygons. Die Fläche des Polygons ergibt sich durch die Summe der Flächen aller Dreiecke $\Delta P P_i P_{i+1}$ zwischen dem inneren Punkt und zwei aufeinander folgenden Grenzpunkten P_i und P_{i+1}:

$$\text{Fläche}_{Polygon} = \sum_{i=1}^{n} \text{Fläche}_{\Delta P P_i P_{i+1}} = \frac{1}{2} \sum_{i=1}^{n} |\det \begin{pmatrix} x & x_i & x_{i+1} \\ y & y_i & y_{i+1} \\ 1 & 1 & 1 \end{pmatrix}|.$$

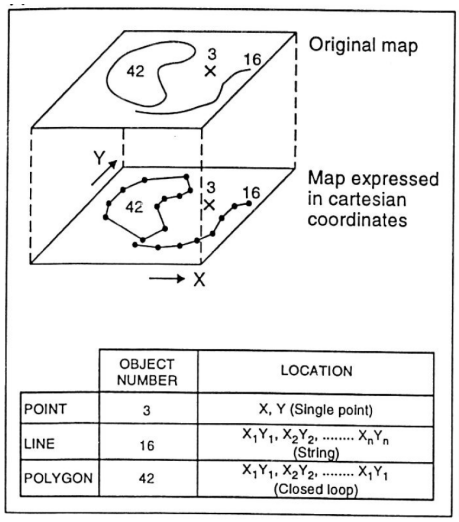
Diese Formel kann zu $\text{Fläche}_{Polygon} = \sum_{i=1}^{n} |x_i \cdot y_{i+1} - x_{i+1} \cdot y_i|$ vereinfacht werden. Zum gleichen Ergebnis kommt man auch ohne die initiale Verwendung eines inneren Punktes aus dem Satz von Green (Green's theorem):

$$\text{Fläche}_{Polygon} = \tfrac{1}{2} |\sum_{i=1}^{n} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \times \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix}|$$

(Quelle: *Graphic Gems II*, James Arvo (ed.), 1991).

Klassischerweise werden Vektorobjekte als Koordinatentabellen gespeichert, in denen für jedes Objekt die Liste der Koordinaten vorgehalten wird. Im einfachsten Fall (siehe Spagettimodell) werden die Koordinaten doppelt genutzter Vertices mehrfach gespeichert. Dies ist zwar recht effizient für die Darstellung, ist aber redundant und erschwert andere Operationen. Diese können durch die Verwendung von topologischen Datenstrukturen effizienter gestaltet werden. In diesen werden die Vertexkoordinaten eindeutig separat gespeichert und über so genannte Inzidenz- oder Verwendungs-Tabellen den sie verwendenden Objekten zugeordnet. Durch die Verwendung von Koordinatenlisten und Objekttabellen lassen sich auch sehr komplizierte Vektorobjekt sehr effizient speichern. Multiple zusätzliche Attribute lassen sich den Objekttabellen als zusätzlich Spalten hinzufügen. Das Speichern von vielen zusätzlichen Variablen an Punkten, Linien und Polygonen ist so sehr einfach möglich. Die allgemeinen Eigenschaften des Vektormodells lassen sich wie folgt zusammenfassen ([Bil10]):

- Basiselemente sind Vertices für Punktobjekte, Linienzüge für Linienobjekte und Polygone für Flächenobjekte
- Linienhafte Betrachtungsweise, Daten zu werden hauptsächlich nach Objektlinien geordnet
- Über Tabellen lassen sich die Objekte und Objektbeziehungen logisch und nachvollziehbar speichern
- Effiziente Speicherung und Darstellung

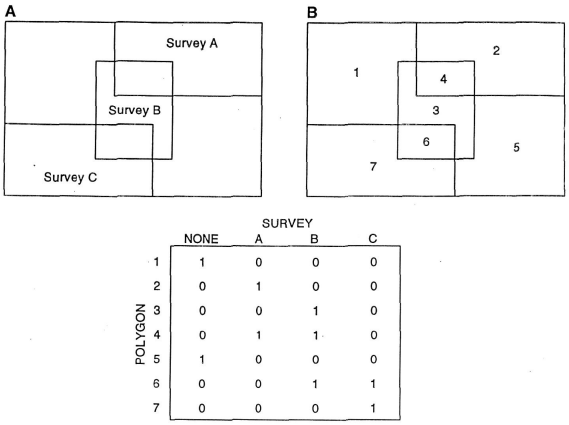


	OBJECT NUMBER	LOCATION
POINT	3	X, Y (Single point)
LINE	16	$X_1Y_1, X_2Y_2, \ldots\ldots X_nY_n$ (String)
POLYGON	42	$X_1Y_1, X_2Y_2, \ldots\ldots X_1Y_1$ (Closed loop)

Abbildung 27: Objekte repräsentiert im Vektormodell als Punkte, Linien und Polygone ([BC94])

planar enforcement Eine Menge von unabhängigen Polygonen in einem Untersuchungsgebiet überdecken dieses nicht notwendiger Weise komplett. Zusätzlich könnten diese Polygone sich noch gegenseitig überlappen. Diese Überlappung können sich negativ auf Auswertung und Darstellung auswirken. Über die Methode des *planar enforcement* ([BC94]) werden diese sich überlappenden Polygone in eine Menge neuer Polygone zerlegt, welche sich nicht überlappen, sondern aneinander grenzen und das Untersuchungsgebiet komplett überdecken. Diese neuen Polygone bilden so eine **polygonale Vermaschung** des Untersuchungsgebietes.

Im unten dargestellten Beispiel für *planar enforcement* bilden drei sich überlappende Polygone (Survey A, B, C) die Ausgangslage. Das Gesamtuntersuchungsgebiet wird durch diese drei Polygone aber nicht komplett überdeckt. In einem ersten Schritt werden die drei Ausgangspolygone so mit dem Polygon des Untersuchungsgebietes und allen vorhandenen Polygonen verschnitten, dass sich 7 unregelmäßig geformte, aneinander grenzenden Polygone bilden. Für jedes dieser Polygone wird ermittelt, welche der initialen Polygonen es überdeckt. Diese Überdeckungen werden in einer Tabelle gespeichert, wodurch sich zum Beispiel sehr einfach das Schnittpolygon zwischen Survey B und Survey C (Polygon 6) abfragen lässt.



POLYGON	SURVEY NONE	A	B	C
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	1	1	0
5	1	0	0	0
6	0	0	1	1
7	0	0	0	1

FIG 2-6. A. A map showing boundaries of three surveys carried out in different years. Note that the survey polygons overlap in some areas, and are completely absent in other areas. In a spaghetti data structure, this is legal. B. The same surveys after **planar enforcement**, forcing the creation of area objects (polygons) that are mutually exclusive and exhaustive. An attribute table is now required to relate polygons to surveys.

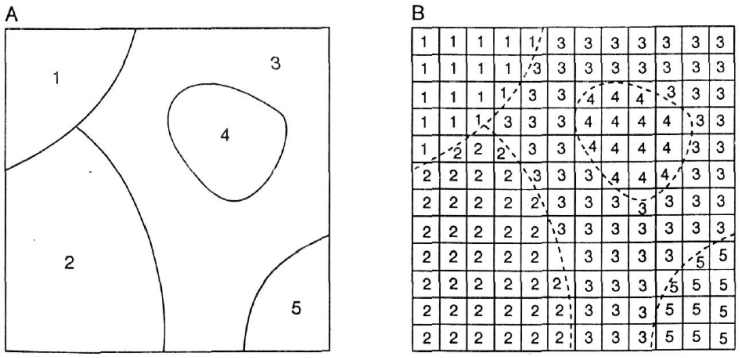
Abbildung 28: planar enforcement ([BC94])

Eine **ebene Vermaschung** ist die Zerlegung eine Flächenobjektes in eine Menge miteinander verbundener, sich nicht überlappender Polygone. *Planar enforcement* ist eine Möglichkeit, eine solche Vermaschung zu erzeugen. Andere Möglichkeiten sind so genannte Triangulationen (Zerlegung in Dreiecke, siehe *Delaunay-Triangulation*) oder Voronoi-Vermaschungen (auch als Thiessen-Polygone /-Vermaschung bezeichnet). Vermaschungen sind vektor-basierte Strukturen, welche sich auch als Graph ihrer Basiselemente repräsentieren lassen. Im Abschnitt *Vermaschungen* wird darauf näher eingegangen.

4.2.3 Attribute und logische Datenmodelle

Geoobjekte lassen sich über ihre Eigenschaften / Attribute beschreiben und in einer Datenbank repräsentieren. Dabei werden die Attribute in **räumliche Attribute** (*spatial*), **zeitliche Attribute** (*temporal*) und **thematische Attribute** (*thematic*) unterteilt. Räumliche Attribute umfassen die Geodaten eines Objektes und beschreiben so Position, Geometrie und Topologie von räumlichen Objekten. Zeitliche Attribute befassen sich mit einem zeitlichen Kontext wie zum Beispiel dem "Alter" eines Objekts, dem Messzeitpunkt oder Zeiträumen. Thematische Attribute umfassen alle nicht-räumlichen oder nicht-zeitlichen Eigenschaften eines Objektes. Temporale und thematische Attribute stellen die Sachdaten oder nicht-räumlichen (*non-spatial*, [BC94]) eines räumlichen Objektes dar. Im GIS-Kontext wird deshalb häufig nicht zwischen ihnen unterschieden. Grafische Attribute (siehe *Geoinformationssysteme*) sind nicht direkt Attribute eines Objektes, sondern Attribute des Darstellungssystems für ein Objekt.

Geoobjekte werden normalerweise über ihre Attribute in Listen oder Tabellen organisiert. Solche Attributtabellen können so als Verbindungsglied zwischen der Vektor- und Rasterrepräsentation eines Objektes angesehen werden. In der folgenden Abbildung beziehen sich sowohl die Vektorrepräsentation (A) als auch die Rasterrepräsentation (B) einer geologischen Karte auf die gleiche Attributtabelle (C). Die Beziehung erfolgt über einen eindeutigen Identifizierer (`Polygon`) in der Tabelle, über den auf die zusätzlichen Informationen für eine Region zugegriffen werden kann.



C

Polygon	Class	Rock type	Age	Name
1	15	sandstone	Late Pennsylvanian	Andrews Formation
2	6	limestone	Early Silurian	Barry Formation
3	3	shale	Middle Silurian	Clinton shale
4	14	granite	Devonian	Delta granite
5	14	granite	Devonian	Delta granite

FIG. 2-11. A. Soil map in vector model. **B.** Same map in and raster model. **C.** Both models utilize the same polygon attribute table. Note that the attribute value in the raster is a pointer to the polygon number.

Abbildung 29: Attributtabelle als Link zwischen Vektor- und Rasterrepräsentation ([BC94])

Die Attributtabellen werden in Datenbanksystemen verwaltet. Dafür müssen die Tabellen gemäß den durch eine Anwendung benötigten Anforderungen strukturiert werden. Ein logisches Datenmodell ([Bil10]) ermöglicht es, die logischen Zusammenhänge zwischen verschiedenen Objekten und ihren Attributen abzubilden. Dafür müssen zuerst die realen Sachverhalte abstrahiert werden.

Entitäten-Relationen-Modell

Reale Objekte und ihre Beziehungen lassen sich zum Beispiel über das so genannten **Entitäten-Relationen-Modell** (*Entity-Relationship-Model, ER-Modell*; [Bil10]) modellieren. Eine **Entität** ist abstraktes Objekt. Entitäten gleichen Typs lassen sich zu Entitätsmengen zusammenfassen. Der Entitäts-Typ wird durch die für diese Objekte definierten Attribute bestimmt. Alle Entitäten einer Menge weisen so die gleichen Attribute (aber ggf. unterschiedliche Attributswerte) auf. Mindestens eines dieser Attribute muss dabei geeignet sein, eine Entität eindeutig zu identifizieren. Zwischen Entitätsmengen werden so genannte Beziehungen oder **Relationen** definiert, welche beschreiben, wie Elemente einer Entitätsmenge mit Elementen einer anderen Entitätsmenge in Verbindung stehen können. Folgenden Beziehungen sind dabei möglich:

1-1-Relation: Eine Entität des Typs A ist umkehrbar eindeutig mit eine einer Entität des Typs B verknüpft.

1-n-Relation: Eine Entität des Typs A ist mit n Entitäten des Typs B verknüpft.

m-n-Relation: m Entitäten des Typs A sind mit n Entitäten des Typs B verknüpft.

Sowohl die Entitätsmengen mit allen Attributen als auch die Relationen werden vor der Modellierung definiert und stehen danach fest.

Um die Bebauung von Flurstücken zu modellieren, benötigt man zum Beispiel zwei Entitätsmengen `Flurstück` (Attribute z.B. `Flurstücks-Nr.` und `Fläche`) und `Gebäude` (Attribute z.B. `Gebäude-Nr.`, `Grundfläche` und `Stockwerke`). Zwischen beiden Mengen wird die Beziehung definiert. Da ein Flurstück sowohl mit mehreren Gebäuden bebaut sein kann, als auch sich ein Gebäude über mehrere Flurstücke erstrecken kann, ist diese Beziehung vom Typ m-n. Diese sehr einfache Modellierung lässt sich über das nachfolgende ER-Diagramm visualisieren. Die Modellierung über Entitäten und Relationen bleibt selbst in der realen Anwendung sehr abstrakt und dient als Grundlage für eine weitergehende logische Datenmodellierung.

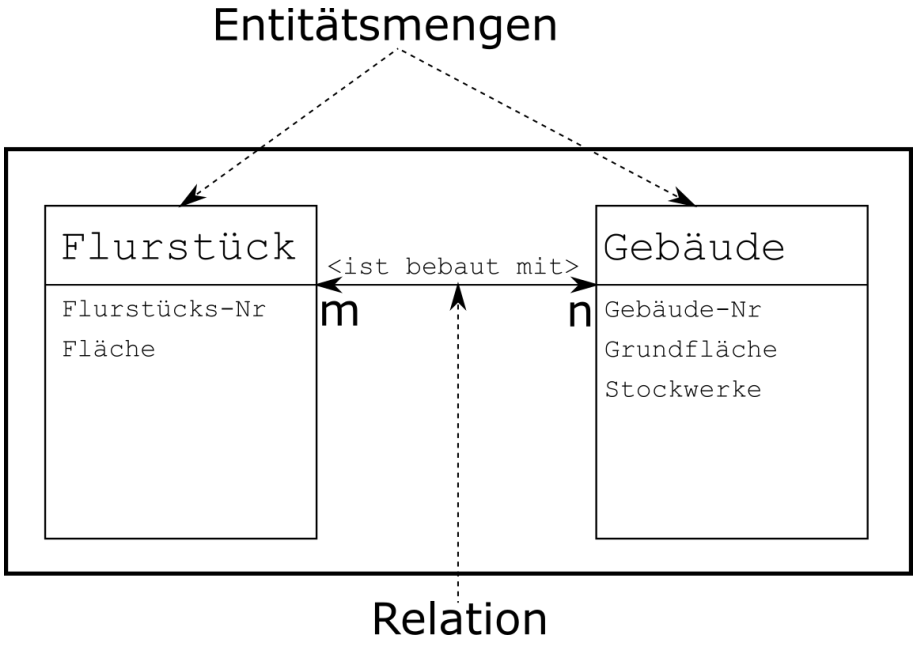


Abbildung 30: Beispielhaftes ER-Diagramm für die Bebauungsbeziehung zwischen Gebäuden und Flurstücken. Die Attribute Flurstücks-Nr. und Gebäude-Nr. dienen der eindeutigen Identifizierung

Graphen-basierte Datenmodelle

Bei Graphen-basierten Datenmodelle lassen sich die Beziehungen zwischen Entitäten als gerichteter Graph darstellen. Die Beziehungen zwischen Entitätsmengen unterscheiden immer zwischen Ausgangsentität und Folgeentität. So sind in der nachfolgenden folgenden Abbildung die Polygone (125 und 126) der Karte K nachgeordnet. Die Reihenfolge der Anhägigkeiten zwischen den Entitätsmengen wird im Voraus festgelegt und bleibt bestehen. Im Sachverhalt aus der nachfolgenden Abbildung sind Polygone immer abhängig von der Karte, Kanten (a bis f) immer abhängig von Polygonen und Vertices (1 bis 6) immer abhängig von Kanten. Die direkte Abhängigkeit zum Beispiel der Punkte von der Karte ist nicht vorgesehen und lässt sich nachträglich nicht ohne sehr großen Aufwand in das Datenmodell integrieren.

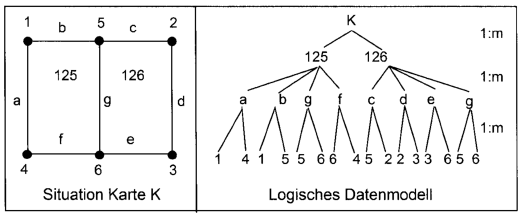


Abbildung 6.43: Hierarchisches Datenmodell.

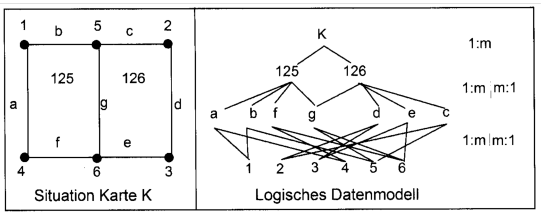


Abbildung 6.44: Netzwerkartiges Datenmodell.

Abbildung 31: Abbildung einer geometrischen Situation aus zwei verbundenen Polygonen mittels Graphen-basierter Datenmodelle ([Bil10])

Hierarchische Datenmodelle setzen für die Beziehung zwischen Entitäten immer eine feste Vater-Sohn-Beziehung voraus. Dies bedeutet, dass eine Ausgangsentität einerseits der Folgeentität übergeordnet ist und andererseits zwar die Vater-Entität mit mehreren Sohn-Entitäten in Beziehung stehen kann, aber eine Sohn-Entität sich nur auf einen Vater beziehen darf. Dies ist eine strikte 1-n-Relation. Fälle, in denen sich ein Sohn auf mehrere Väter"beziehen soll, können nur durch redundantes Vorhalten dieser Sohn-Entität abgebildet werden. Dies ist in der oberen Abbildung (links) zum Beispiel für Kante g der Fall, welche von beiden Polygonen verwendet werden soll. Sachverhalte, die eine solche stark hierarchische Struktur aufweisen, lassen sich mit hierarchischen Datenmodellen sehr gut abbilden. Das Ergebnis ist eine Baumstruktur, für die sehr effiziente Verarbeitungs- und Verwaltungsmethoden existieren. Häufig lassen sich jedoch reale Sachverhalte nicht ausschließlich über 1-n-Relationen in einer festen Reihenfolge adäquat abbilden.

Eine Weiterentwicklung der hierarchischen Datenmodelle sind *Netzwerk-Datenmodelle* (obere Abbildung, rechts). Diese erlauben echte m-n-Relationen. Es wird zwischen owner- und member-Entitäten unterschieden. Eine owner-Entität (Ausgangsentität) darf sich auf beliebig viele member-Entitäten (Folgeentität), ein member aber auch auf mehrere owner beziehen. Statt über einen Beziehungsbaum werden die Beziehungen jetzt über ein Beziehungsnetz abgebildet. Auch für solche Netzwerke existieren effiziente Verarbeitungsmethoden. Allerdings ist der Aufbau und die Veränderung solcher Netzwerke oder Graphen zum Teil sehr aufwändig. Der abzubildende Sachverhalt sollte also nicht zu groß und möglichst statisch sein, d.h. sich nachträglich nicht oder kaum ändern. Datenmodelle dieser Art werden häufig für die Verwaltung von Geometrie- und Topologiedaten innerhalb von GIS verwendet.

Graphen-basierte Datenmodelle gehören zu den ältesten verwendeten Datenmodellen. Sie werden aber heutzutage immer noch genau in den Fällen bevorzugt, in denen der abzubildende Sachverhalt diese Art der Modellierung erlaubt, da die Verarbeitung sehr effizient möglich ist. In der Regel werden für die Umsetzung von Entitäten und Beziehungen für diese Art der Datenmodelle unterschiedliche Datenstrukturen verwendet.

Relationales Datenmodell

Im relationalen Datenmodell werden sowohl Entitäten als auch Beziehungen in tabellarischer Form verwaltet. Dabei ist die Tabelle (bezeichnet als **Relation**) die einzig gültige Datenstruktur. Eine strikte Unterscheidung zwischen Ausgangs- und Folgeentitäten einer Beziehung und auch eine feste Beziehungsabfolge werden nicht vorausgesetzt.

Jede Tabellen-Zeile einer Relation ist ein so genanntes **Tupel**, ein geordneter Satz von Werten für die verschiedenen Attribute. Ein **Attribut** entspricht einer Tabellenspalte und definiert ein Merkmal einer Entität oder Relation. Attributwerte können entweder kontinuierlich sein, d.h. unendlich viele verschiedene Werte annehmen, oder diskret sein. Diskrete Attribute dürfen nur Werte aus einem festgelegten Wertebereich (genannt **Domäne**) annehmen. Attribute oder Attributkombinationen, welche zur Identifizierung einzelner Tupel oder Entitäten verwendet werden, werden als **Schlüssel**-Attribute (*key*) bezeichnet. **Primärschlüssel** identifizieren die Tupel innerhalb einer Relation eindeutig, d. h. die Attributwerte dürfen in dieser Relation nur einmal auftreten. **Fremdschlüssel** verweisen auf Tupel einer anderen Relationen und dürfen innerhalb der verweisenden Relation nicht eindeutig sein, d.h. mehrere Tupel dürfen auf das gleiche Tupel einer anderen Relation verweisen. Jede Relation muss dabei eindeutig identifizierbar sein, z.B. über einen eindeutigen Namen oder Nummerierung verfügen. Im Allgemeinen lässt sich jede Relation über

```
Relationsname(Schlüssel-Attribut 1, Schlüssel-Attribut 2, ... ,
                    Attribut 1, Attribut 2,...)
```

beschreiben (Notation nach [BC94]).

Grundsätzlich dürfen Relationen beliebige komplex sein, solange sie folgende Bedingungen erfüllen:

1. Tupel dürfen sich innerhalb einer Relation nicht wiederholen.
2. Die Bedeutung einer Relation darf nicht von der Reihenfolge der Tupel oder Attribute abhängen. Das heißt, dass es möglich sein muss, sowohl die Tupel als auch die Attribute einer Relation untereinander zu vertauschen, ohne dass sich die Bedeutung verändert.

Eine diesbezüglich valide Relation wäre zum Beispiel die folgende Relation `Polygone` (Beisiel basiert auf [BC94]). In ihre werden verschiedenen Polygone geologische Formationen zugeordnet. Diese Relation kann über `Polygone(Polygon #, Formation, Lithologie, Alter, Zeitspanne)` beschrieben werden. Die erste Zeile beinhaltet den Relationsnamen, die Zweite die Namen der Attribute. Das Attribut `Polygon #` ist der **Primärschlüssel**.

Tabelle 2: Ursprüngliche Beispielrelation.

Polygone				
Polygon #	**Formation**	**Lithologie**	**Alter**	**Zeitspanne**
1	Shelly Fm.	Kalkstein	Oberkarbon	323.2 – 298.9 MJ
2	Grit Fm.	Sandstein	Oberkarbon	323.2 – 298.9 MJ
3	Slab Fm.	Schiefer	Oberkarbon	323.2 – 298.9 MJ
4	Mount Fm.	Granit	Kreide	145 – 66 MJ
5	Mount Fm.	Granit	Kreide	145 – 66 MJ
6	Volcano Fm.	Tuff	Trias	251.9 – 201.3 MJ
7	Mount Fm.	Granit	Kreide	145 – 66 MJ
8	Shelly Fm.	Kalkstein	Oberkarbon	323.2 – 298.9 MJ
9	Slab Fm.	Schiefer	Oberkarbon	323.2 – 298.9 MJ
10	Shelly Fm.	Kalkstein	Oberkarbon	323.2 – 298.9 MJ

Beliebig komplexe Relationen weisen häufig das Problem der Redundanz auf und lassen sich deshalb nur aufwändig erweitern oder ändern. Um dies zu erleichtern, ist es möglich, komplexe Relationen in einfachere, auf einander verweisende Relationen aufzuspalten. Dieser Vorgang wird als **Normalisierung** bezeichnet. In einem ersten Schritt der Normalisierung werden numerische Codes für text-basierte Attribute eingeführt. Dadurch lassen sich später Probleme durch Bedeutungsänderungen oder verschiedene Schreibweisen umgehen. Für die oben gezeigte Polygonrelation könnte dies zu folgendem Ergebnis führen:

Tabelle 3: Beispielrelation mit numerischen Codes.

Polygone							
Polygon #	**Formation #**	**Formation**	**Lithologie #**	**Lithologie**	**Alter #**	**Alter**	**Zeitspanne**
1	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 – 298.9 MJ
2	3	Grit Fm.	6	Sandstein	5	Oberkarbon	323.2 – 298.9 MJ
3	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 – 298.9 MJ
4	1	Mount Fm.	2	Granit	8	Kreide	145 – 66 MJ
5	1	Mount Fm.	2	Granit	8	Kreide	145 – 66 MJ
6	5	Volcano Fm.	3	Tuff	7	Trias	251.9 – 201.3 MJ
7	1	Mount Fm.	2	Granit	8	Kreide	145 – 66 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 – 298.9 MJ
9	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 – 298.9 MJ
10	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 – 298.9 MJ

Es ist leicht zu sehen, dass diese Relation Redundanzen enthält. Bestimmte Attributwerte für `Formation` treten mehrfach auf. Die Attribute `Lithologie`, `Alter` und `Zeitspanne` scheinen sich auf Formation beziehen, da ihre Werte immer dann wiederholt auftreten, wenn sich auch der Wert für die Formation wiederholt. Durch die Normalisierung lässt sich diese Relation jetzt nach bestimmten Regeln in so genannte Normalformen überführen, wodurch sich ihre Struktur vereinheitlichen und Redundanzen entfernen lassen.

Für die ***1. Normalform*** (1NF) muss die Relation folgende Bedingungen erfüllen:

- Alle Attributwerte müssen **atomar** sein, das heißt, sie dürfen nur aus einem einzigen Wert bestehen.
- Die Relation muss **frei von Wiederholungsgruppen** sein. Wiederholungsgruppen sind Attribute mit gleicher oder gleichartiger Bedeutung. Ausnahme hier sind die numerischen Codes für text-basierte Attribute.
  Oder Alternativ: **Die Anzahl der Attributwerte pro Tupel muss gleich sein**; die Relation muss "eine konstante Breite" aufweisen.

In der Beispielrelation sind alle Attribute bis auf die Zeitspanne bereits atomar. Das Attribut `Zeitspanne` enthält zwei Werte. Auch wenn die Zeichenketten für die Formation Leerzeichen enthalten und somit aus mehreren Worten bestehen, so ist doch die gesamt Formationsbezeichnung als zusammengehöriger Attributwert zu sehen. Um diese Relation in die 1. Normalform zu überführen, muss zuerst das Attribut `Zeitspanne` atomisiert, das heißt in zwei atomare Attribute `Beginn Zeitspanne` und `Ende Zeitspanne` aufgespalten werden.

Wiederholungsgruppen liegen in der Beispielrelation nicht vor. Eine Wiederholungsgruppe wäre zum Beispiel, wenn man verschiedene Gemeinden einem Land zuordnen möchte und für eine Land-Entität mehrere Attribute Gemeinde für verschiedene Gemeindenamen definiert wären. Dies könnte dazu führen, dass verschiedene Entitäten für Länder unterschiedlich viele Attribute für die Gemeinden aufweisen. Dies wird über die Auslagerung der Gemeinde-Landbeziehung in eine separate Relation, in der jeder Gemeindename einem Land zugeordnet wird, vermieden. Relationen, die der 1. Normalform entsprechen, müssen für jedes Tupel die gleiche Anzahl von Attributen aufweisen ([Bar05]).

Die Beispielrelation könnte dann in der 1. Normalform wie folgt aussehen. Sie beinhaltet weiterhin redundante Informationen.

Tabelle 4: Beispielrelation in 1NF.

Polygone								
Polygon #	**Formation #**	**Formation**	**Lithologie #**	**Lithologie**	**Alter #**	**Alter**	**Beginn Zeitspanne**	**Ende Zeitspanne**
1	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
2	3	Grit Fm.	6	Sandstein	5	Oberkarbon	323.2 MJ	298.9 MJ
3	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
4	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
5	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
6	5	Volcano Fm.	3	Tuff	7	Trias	251.9 MJ	201.3 MJ
7	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
9	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
10	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ

Nach der Normalisierung in der 1NF ist die Beschreibung der Relation

```
Polygone(Polygon #, Formation #, Formation, Lithologie #, Lithologie, ...
 Alter#, Alter, Beginn Zeitspanne, Ende Zeitspanne).
```

Für die **2. Normalform** (2NF) muss eine Relation folgende Bedingungen erfüllen:

- **Die Relation muss in der 1. Normalform vorliegen.**
- **Jedes Nicht-Schlüssel-Attribut muss von der gesamten Schlüsselkombination abhängen.**

Bedingung 2 ist immer dann relevant, wenn der Schlüssel aus mehreren Attributen zusammengesetzt ist. Dann muss jedes Attribut, dass kein Schlüsselattribut ist, immer von allen Schlüsselattributen abhängen und nicht nur von einer Untermenge. Angenommen es liegt folgende Relation vor:
`Relation(Key 1, Key 2, Attr. 1, Attr. 2, Attr. 3)`.
`Attr. 1` und `Attr. 2` hängen von der Kombination aus `Key 1` und `Key 2` ab, `Attr. 3` allerdings nur von `Key 1`. Zum Erreichen der 2NF müsste der Verweis auf `Attr. 3` aus Relation entfernt und in eine neue Relation überführt werden. Danach liegen zwei Relationen
`Relation(Key 1, Key 2, Attr. 1, Attr. 2)` und `Relation2(Key 1, Attr. 3)` vor.
Da die Beispielrelation Polygone nur ein Schlüsselattribut aufweist (`Polygon #`), liegt die in die 1. Normalform normalisierte Relation auch automatisch in der 2. Normalform vor.

Für die **3. Normalform** (3NF) muss eine Relation folgende Bedingungen erfüllen:

- **Die Relation muss in der 2. Normalform vorliegen.**
- **Kein Nicht-Schlüssel-Attribut darf *transitiv* von den Schlüsselattributen abhängen.**

Transitiv bedeutet hier, dass die Abhängigkeit vom Schlüssel nur indirekt ist. Ein transitiv abhängiges Attribut hängt direkt von einem anderen Nichtschlüssel-Attribut ab, welches direkt vom Schlüssel abhängt. Diese transitiven Abhängigkeiten müssen in gesonderte Relationen ausgelagert werden.

Für die Beispielrelation ist deutlich zu erkennen, dass die Attribute für Alter und Lithologie nicht direkt vom Polygon-Schlüssel, sondern von den Formationsattributen abhängen. Für die Normalisierung in die 3NF müssen deshalb die Formationsinformationen von den Polygoninformationen entkoppelt werden. Dies vereinfacht die Polygonrelation zu `Polygone(Polygon #, Formation #)` mit nur noch einem Nicht-Schlüssel-Attribut für die 10 Polygon-Entitäten.

Tabelle 5: Beispielrelation Polygone in 3NF.

Polygone	
Polygon #	**Formation #**
1	2
2	3
3	4
4	1
5	1
6	5
7	1
8	2
9	4
10	2

Zusätzlich wird eine zweite Relation
`Formation(Formation #,Formation, Lithologie #, Lithologie, ...`
`... Alter#, Alter, Beginn Zeitspanne, Ende Zeitspanne)`
aufgebaut, welche nur 5 Tupel für die Formations-Entitäten enthält. In dieser Relation lassen sich noch die Lithologie- und Alter-Informationen von der Formations-Information entkoppeln. Dadurch entstehen folgende Relationen:
`Formation(Formation #,Formation, Lithologie #, Alter#)`
`Lithologie(Lithologie #, Lithologie)`
`Alter(Alter#, Alter, Beginn Zeitspanne, Ende Zeitspanne)`

Tabelle 6: Beispielrelation Formation in 3NF.

Formation			
Formation #	**Formation**	**Litholohie #**	**Alter #**
1	Mount Fm.	2	8
2	Shelly Fm.	7	5
3	Grit Fm.	6	5
4	Slab Fm.	5	5
5	Volcano Fm.	3	7

Tabelle 7: Beispielrelation Lithologie in 3NF.

Lithologie	
Litholohie #	**Lithologie**
2	Granit
3	Tuff
5	Schiefer
6	Sandstein
7	Kalkstein

Tabelle 8: Beispielrelation Alter in 3NF.

Alter			
Alter #	**Alter**	**Beginn Zeitspanne**	**Ende Zeitspanne**
5	Oberkarbon	323.2 MJ	298.9 MJ
7	Trias	251.9 MJ	201.3 MJ
8	Kreide	145 MJ	66 MJ

Der abgebildete relationale Sachverhalt liegt jetzt in der 3NF ohne Redundanzen vor.

Grundsätzlich gibt es die Möglichkeit der Normalisierung in weitere Normalformen, z. B. **Boyce-Codd-Normalform** (BCNF), **4. Normalform** (4NF), **5. Normalform** (5 NF) usw. Dies ist allerdings nicht immer verlustfrei möglich. Relationen in der 3NF erlauben allerdings bereits die effiziente Verarbeitung eines relationalen Sachverhalts mittels Operationen der relationalen Algebra. Diese sind nach Bartelme (2005, [Bar05]):

- Vereinigung von Relationen,
- Differenz von Relationen,
- Durchschnitt von Relationen,
- Projektion,
- Selektion,
- kartesisches Produkt.

Bei der **Vereinigung** zweier Relationen A und B wird eine neue Relation C aufgebaut, welche alle Tupel aus den beiden Ursprungsrelationen enthält. Treten Tupel mehrfach auf, werden die Duplikate entfernt. Die **Differenz** zweier Relationen A und B ergibt eine neue Relation C, welche nur die Tupel von A enthält, welche nicht auch in B auftreten. Der **Durchschnitt** von A und B enthält nur Tupel, welche sowohl in A als auch in B auftreten. Diese drei Operationen sind nur für Relationen gleichen Typs möglich, das heißt für Relationen, welche die gleichen Attribute aufweisen.

Die **Projektion** erlaubt die Auswahl einzelner Spalten einer Relation, die **Selektion** erlaubt die Auswahl einzelner Tupel, welche eine gegebene Bedingung hinsichtlich ihrer Attributwerte erfüllen. Beides sind wichtige Elementaroperationen für nicht-räumliche Abfragen in einem GIS.

Das **kartesische Produkt** assoziiert jedes Tupel einer Relation A (n Tupel) mit jedem Tupel einer Relation B (m Tupel) in einer neuen Relation. Ausgehend von zwei Relation `A(keyA, A)` und `B(keyB, B)` wird durch das kartesische Produkt die Relation `C(keyA, A, keyB, B)` erzeugt, welche wie folgt aussehen könnte:

Tabelle 9: Beispielrelation für das Ergebnis eines kartesischen Produktes.

C			
keyA	**A**	**keyB**	**B**
a1	Wert A1	b1	Wert B1
a1	Wert A1	b2	Wert B2
⋮	⋮	⋮	⋮
a1	Wert A1	bm	Wert Bm
a2	Wert A2	b1	Wert B1
⋮	⋮	⋮	⋮
an	Wert An	bm	Wert Bm

Über das relationale Datenmodell lassen sich auch komplexe Sachverhalte sehr strukturiert speichern. Komplexe räumliche Abfragen sind aufgrund der starken Fragmentierung in viele elementare Relationen allerdings sehr zeitaufwändig. Zudem ist es schwierig, komplette Objekte als eine Gesamtmenge zusammengehöriger Attributwerten zu modellieren. Dafür bieten sich statt dessen Objekt-basierte Datenmodelle an.

Objekt-basiert Datenmodelle

Objekte-basierte Datenmodelle orientieren sich an den Mitteln der Objekt-orientierten Programmierung. Entitäten (**Objekte**) werden immer als zusammengehörige, nicht zu trennende Menge von Attributwerten betrachtet. Sie sind Realisierungen (**Instanzen**) von **Klassen**. Diese Klassen entsprechen den Entitätstypen und definieren sowohl die Attribute ihrer Objekte, als auch deren mögliche Operationen (**Methoden**). Über **Vererbung** und **Polymorphie** können neue Klassen aus vorhanden Klassen abgeleitet werden. Objekte können als atomar betrachtet werden und dürften so als Attributwert in einem relationalem Datenmodell verwendet werden (**Objekt-relationales-Datenmodell**). Es existieren zudem noch andere Arten von objekt-basierten Datenmodellen.

Die Vorteile des Objekt-basierten Ansatzes liegen darin, dass sich auch komplexe reale Sachverhalte leicht und flexibel als Objekte abbilden lassen. Eine nachträglicher Erweiterung von Objekten um neue Attribute/Methoden ist leicht möglich, da diese Änderung nur in der Klasse ausgeführt werden muss und nicht für jedes Objekt separat. Zudem kann jedes Objekt seine Daten selbst verwalten. Der Nachteil ist, dass, anders als bei relationalen Datenbanken, noch kein allgemein gültiger Standard für objekt-orientierte Datenbanken existiert. Zusätzlich sind objekt-orientierte Datenmodelle typischerweise sehr viel speicher- und rechnen-aufwändiger umzusetzen als das relationale Datenmodell.

4.3 Datenstrukturen

Räumliche Datenstrukturen organisieren die Daten in verschiedenen räumlichen Datenmodellen so, dass sie von einem Computer verarbeitet werden können. Es gibt verschiedene Datenstrukturen für Vektor- und Rasterdaten. Für Vermaschungen, einer speziellen Unterkategorie von Vektordaten, gibt es, zusätzlich zu klassischen Vektor-Datenstrukturen, spezielle Datenstrukturen, um die Vermaschungen effizient darstellen und verarbeiten zu können. Je nach Anwendungsanforderungen gibt es verschiedene Datenstrukturen für Daten in einem Datenmodell.

4.3.1 Datenstrukturen für Rasterdaten

Scan-order zur Speicherung von Rasterdaten

Klassischerweise werden Daten im Rastermodell als Matrizen der Pixelattributwerte gespeichert. Für jedes Attribut (als als Band bezeichnet) wird dabei eine separate Matrix für die Pixelwerte verwendet. Dies bis schematisch in der folgenden Abbildung für zwei Bänder (A und B) dargestellt. Dies Speicherung solcher zweidimensionaler Matrizensysteme ist allerdings vergleichsweise

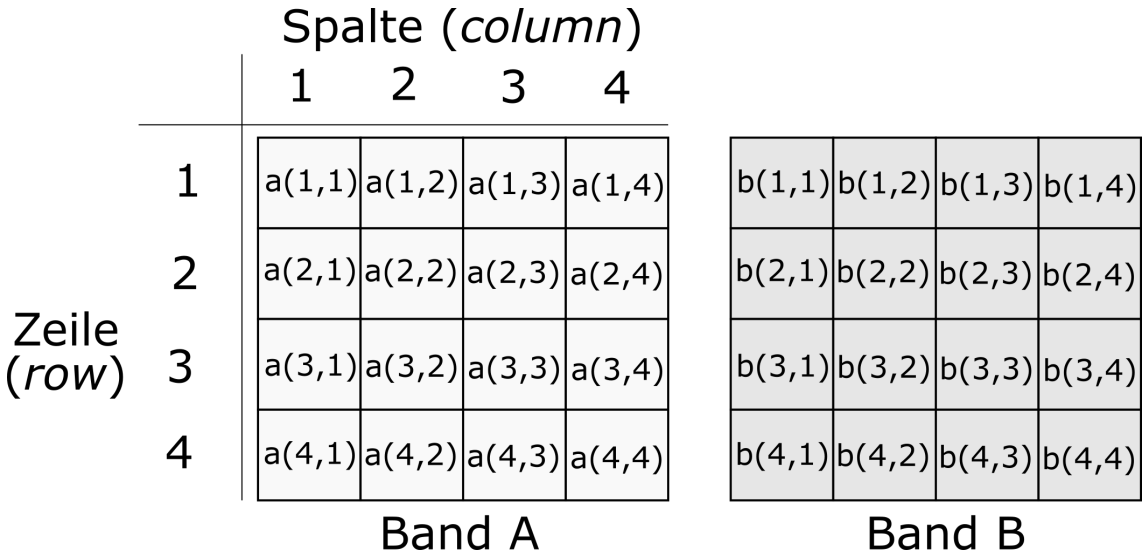


Abbildung 32: Zweidimensionales Beispielraster mit 2 Bändern A und B

aufwändig, wohingegen sich eindimensionale Listen sehr effizient speichern lassen. Das Ziel für die Speicherung von Rasterdaten ist es also, die Pixelmatrizen für alle Bänder eines Rasterdatensatzes in **einer** langen Liste zu speichern. Dafür werden die Pixelwerte gemäß einer vordefinierten Ordnung (***scan-order***) ausgehend vom ersten Pixel (erste Zeile, erste Spalte) als Liste aneinandergereiht. In der nachfolgenden Abbildung sind verschiedene Möglichkeiten für solche *scan-orders* nach Bonham-Carter (1994,[BC94]) dargestellt.

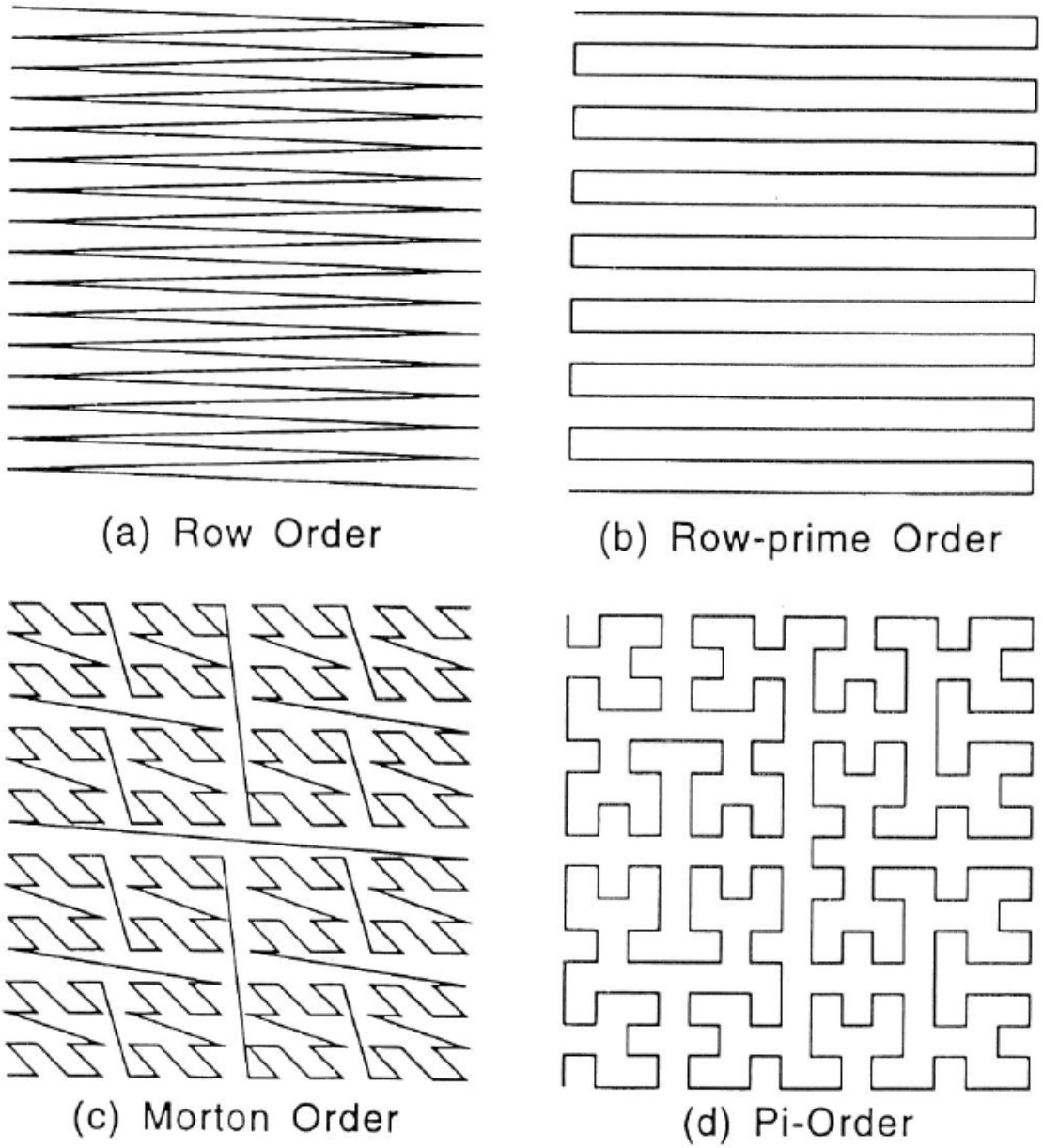


FIG. 3-3. Four different kinds of raster ordering. **a**. Row order (this is the most comon method). **b**. Row-prime order. **c**. Morton order (used in quadtrees). **d**. Hilbert-Peano order. Figure adapted from Goodchild and Grandfield (1983).

Abbildung 33: Schematische Darstellung verschiedener scan-orders nach Bonham-Carter (1994,[BC94])

Im einfachsten Fall, der so genannten **row-order**, werden zuerst alle Pixelwerte der ersten Zeile (*row*) aufgelistet. Dies wird für alle folgenden Zeilen wiederholt. Für Band A des obigen Beispielrasters wäre die Pixelabfolge in der Liste wie folgt:

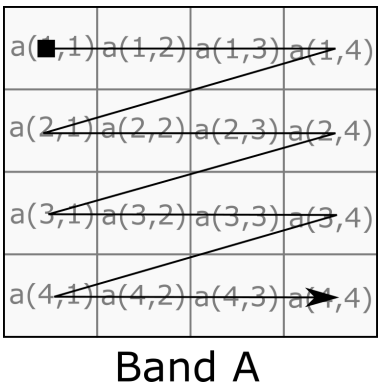


Abbildung 34: Row-order für Band A des Beispiels

Die entstehende Werteliste wäre:
$a(1,1); a(1,2); a(1,3); a(1,4); a(2,1); a(2,2); a(2,3); a(2,4); \ldots$
$\ldots a(3,1); a(3,2); a(3,3); a(3,4); a(4,1); a(4,2); a(4,3); a(4,4).$

Die row-order ist sehr weit verbreitet, hat aber einen praktischen Nachteil. In der Liste stehen der letzte Pixel einer Zeile und der erste Pixel der nächsten Zeile direkt nebeneinander, obwohl sie gemäß der Gitterstruktur sehr weit voneinander entfernt liegen. Dies kann sich nachteilig auf eine spätere Komprimierung oder Verarbeitung der Liste auswirken. Bei der **row-prime-order** wird dieses Problem umgangen, indem auf den letzten Pixelwert einer Zeile der letzte Pixelwert der nachfolgenden Zeile folgt, welche dann in der entgegen gesetzten Richtung abgelaufen wird. Dadurch sind alle aufeinanderfolgenden Pixelwerte in der Liste auch "nah" bezüglich der Gitterstruktur. Für Band A des obigen Beispielrasters wäre die row-prime Pixelabfolge in der Liste wie folgt:

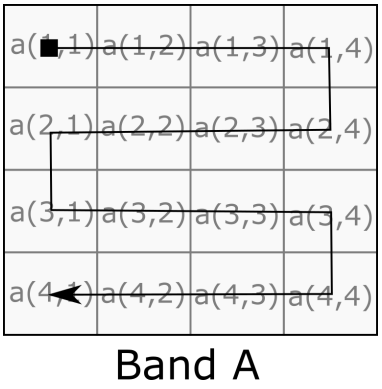


Abbildung 35: Row-prime-order für Band A des Beispiels

Die entstehende Werteliste wäre:
$a(1,1); a(1,2); a(1,3); a(1,4); a(2,4); a(2,3); a(2,2); a(2,1); \ldots$
$\ldots a(3,1); a(3,2); a(3,3); a(3,4); a(4,4); a(4,3); a(4,2); a(4,1).$

So genannte **raum-füllende** (*space-filling*) Pixelabfolgen, wie zum Beispiel die **Morton**-oder die **Hilbert-Peano**-Abfolge, sind nicht mehr zeilen-zentriert, sondern es werden die Pixel blockweise abgelaufen. Dadurch wird sicher gestellt, dass Pixelwerte, welche in der Liste "nah" sind, immer auch gemäß der Gitterstruktur "nah" sind, das heißt, sich im gleichen oder einem benachbarten Gitterblock befinden.

Für Band A des obigen Beispielrasters wäre die Morton-Pixelabfolge in der Liste wie folgt:

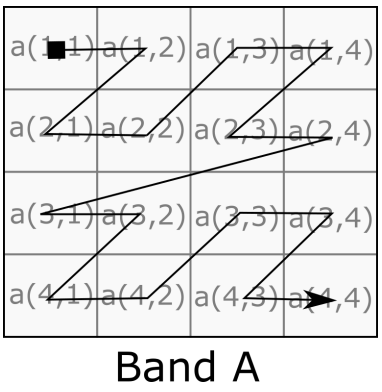


Abbildung 36: Morton-order für Band A des Beispiels

Die entstehende Werteliste wäre:
$a(1,1); a(1,2); a(2,1); a(2,2); a(1,3); a(1,4); a(2,3); a(2,4); \ldots$
$\ldots a(3,1); a(3,2); a(4,1); a(4,2); a(3,3); a(3,4); a(4,3); a(4,4).$

Für Band A des obigen Beispielrasters wäre die Hilbert-Peano-Pixelabfolge (auch als pi-order bezeichnet) in der Liste wie folgt:

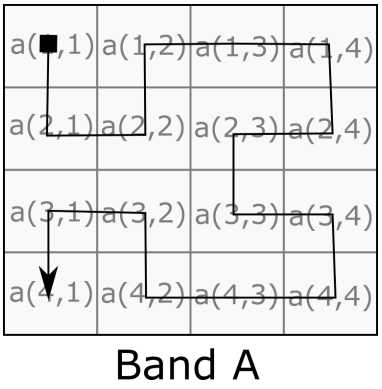


Abbildung 37: Pi-order für Band A des Beispiels

Die entstehende Werteliste wäre:
$a(1,1); a(2,1); a(1,2); a(2,2); a(1,3); a(1,4); a(2,4); a(2,3); \ldots$
$\ldots a(3,3); a(3,4); a(4,4); a(4,3); a(4,2); a(3,2); a(3,1); a(4,1).$

Liegen mehr als ein Rasterband vor, gibt es mehrere Möglichkeiten, die verschiedenen Pixelwerte der verschiedenen Bänder in der Liste anzuordnen. Eine Möglichkeit ist die ***band squential*** (**BSQ**) Anordnung, bei der die Bänder sequentiell hintereinander angeordnet werden. Dies bedeutet, zuerst werden alle Pixelwerte des ersten Bandes gemäß einer gegebenen scan-order aufgelistet. Danach folgen alle Pixelwerte des zweiten Bandes gemäß der gleichen Ordnung. Dies wird für alle Bänder analog durchgeführt. In der Liste stehen so die Pixelwerte eines Bandes "nah" beieinander. Dies ist effizient, wenn die Bänder unabhängig von einander ausgewertet werden sollen.

Bei der ***band interleaved by line*** (**BIL**) Anordnung werden die Bänder zeilenweise hintereinander in die Liste geschrieben. Auf alle Pixelwerte des ersten Bandes einer Zeile folgen die Pixelwertewerte des zweiten Bandes dieser Zeile usw. Erst wenn alle Werte aller Bänder für eine Zeile abgearbeitet sind, wird dies für die folgende Zeile wiederholt. Diese Anordnung kann nur für zeilen-basierte scan-orders angewandt werden, bietet aber einen guten Kompromiss bezüglich der Verarbeitungseffizienz, wenn sowohl die verschiedenen Bänder einzeln als auch kombiniert verarbeitet werden sollen.

Bei der ***band interleaved by pixel*** (**BIP**) Anordnung werden die Bänder pixelweise hintereinander in die Liste geschrieben. Erst werden alle Werte für alle Bänder eines Pixels in die Liste geschrieben, dann alle Werte für das Folgepixel und so weiter. Dadurch lassen sich die verschiedenen Bänder sehr effizient miteinander kombiniert auswerten.

Für das oben gezeigte Beispiel mit zwei Bändern A und B würden die Listen der Pixelwerte ausgehend von einer klassischen row-order wie folgt aussehen:
BSQ: $a(1,1); a(1,2); a(1,3); a(1,4); a(2,1); a(2,2); a(2,3); a(2,4); \ldots$
$\ldots a(3,1); a(3,2); a(3,3); a(3,4); a(4,1); a(4,2); a(4,3); a(4,4); b(1,1); b(1,2); \ldots$
$\ldots b(1,3); b(1,4); b(2,1); b(2,2); b(2,3); b(2,4); b(3,1); b(3,2); b(3,3); b(3,4); \ldots$
$\ldots b(4,1); b(4,2); b(4,3); b(4,4)$
BIL: $a(1,1); a(1,2); a(1,3); a(1,4); b(1,1); b(1,2); b(1,3); b(1,4); \ldots$
$\ldots a(2,1); a(2,2); a(2,3); a(2,4); b(2,1); b(2,2); b(2,3); b(2,4); a(3,1); a(3,2); \ldots$
$\ldots a(3,3); a(3,4); b(3,1); b(3,2); b(3,3); b(3,4); a(4,1); a(4,2); a(4,3); a(4,4); \ldots$
$\ldots b(4,1); b(4,2); b(4,3); b(4,4)$
BIP: $a(1,1); b(1,1); a(1,2); b(1,2); a(1,3); b(1,3); a(1,4); b(1,4); \ldots$
$\ldots a(2,1); b(2,1); a(2,2); b(2,2); a(2,3); b(2,3); a(2,4); b(2,4); a(3,1); b(3,1); \ldots$
$\ldots a(3,2); b(3,2); a(3,3); b(3,3); a(3,4); b(3,4); a(4,1); b(4,1); a(4,2); b(4,2); \ldots$
$\ldots a(4,3); b(4,3); a(4,4); b(4,4)$

Lauf-Längen Kodierung (*run-length encoding*)

Die Speicherung aller Pixelwerte ist vor allem für große Rasterdaten sehr speicher aufwändig. Über eine so genannte **Lauf-Längen-Kodierung** (*run-length encoding*) lassen sich Rasterdaten komprimieren, um den benötigten Speicheraufwand zu reduzieren. Häufig weisen Rasterdaten nur eine begrenzte Menge verschiedener Attributwerte auf. Bei der Komprimierung wird ausgenutzt, dass aufeinanderfolgende Pixel oft gleiche Werte aufweisen. Anstelle von n aufeinanderfolgenden, gleichen Werten a $(a, a, a, \ldots, a)$ zu speichern, wird nur die Lauflänge n und einmal der Wert a gespeichert.

Die Lauf-Längen-Kodierung einer Liste/Zeile könnte wie folgt aussehen:
$a, a, a, b, b, a, a, a, c, c, a, a, a, a, a, b, b, b \rightarrow 3a, 2b, 3a, 2c, 5a, 3b.$

Im Fall einer Pixelmatrix wird die Kodierung zeilen-weise durchgeführt, für jede Matrixzeile wird eine neue kodierte Zeile erstellt.

Eine Lauf-Längen-Kodierung ist besonders effizient, wenn die Pixelliste oder jede Matrixzeile nur wenige verschiedenen Attributwerte enthält. Wenn man annimmt, dass Pixel, welche im Gitter "nah" beieinanderliegen, häufiger gleicher Werte aufweisen, als Pixel, welche "weit" voneinander entfernt sind, lassen sich vor allem raum-füllende Pixel-Abfolgen durch Lauf-Längen-Kodierung sehr effizient komprimieren. Es werden im besten Fall nicht nur weniger Werte gespeichert, als wenn man alle Pixelwerte speichern würde, sondern zusätzlich ist der Speicheraufwand für die Speicherung der Lauf-Länge n zumeist geringer als für einen beliebigen Attributwert. Es ist häufig ausreichend, die Lauf-Länge als Ganzzahl ohne Vorzeichen (unsigned integer) anstelle als Fließkomma-Zahl (*float, double*) für die Attributwerte zu speichern.

Quadtrees und Morton-Koordinaten

Bei einem Quadtree handelt es sich um eine hierarchische Datenstruktur. Diese basiert auf der sukzessiven Zerlegung eines Blockes von Pixeln in 4 gleich große Quadranten. Ein Pixel-Block wird immer dann nicht zerlegt, wenn er eine der folgenden Bedingungen erfüllt:

- Er ist **homogen**, das heißt alle Pixelwerte aller Pixel im Block sind identisch, oder
- er besteht aus genau einem Pixel und ist damit automatisch homogen.

Diese Zerlegung lässt sich auch als Graph darstellen (siehe nachfolgende Abbildung). Der Ausgangsknoten (*root*) umfasst das gesamte Raster. Dieses wird in 4 gleich große Blöcke aufgeteilt, jeder Block entspricht einen Knoten im Graph. Ist ein Block homogen, wird er nicht weiter aufgeteilt und ist so ein so genannter Blatt-Knoten im Graph, also ein Knoten ohne Kind-Knoten. Dies trifft zum Beispiel auf die Knoten 100, 210 und 341 in der nachfolgenden Abbildung zu.

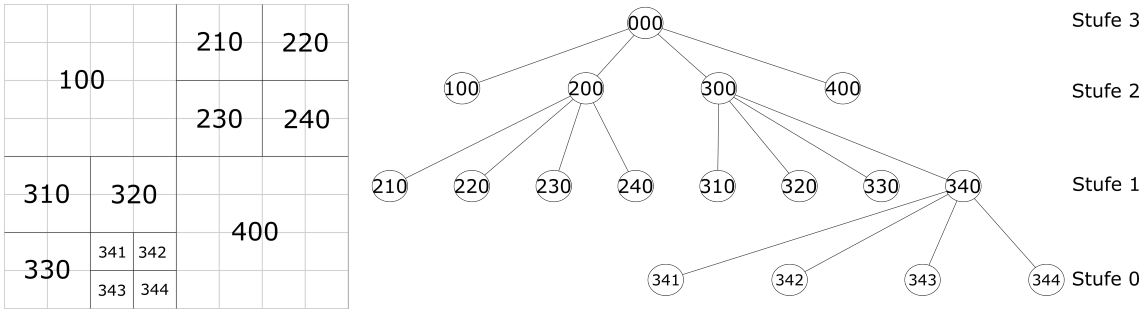


Abbildung 38: Schematische Zerlegung eines 8 x 8 Pixel Rasters in 3 Stufen und die zugehörigen Morton-Koordinaten

Die durch eine Zerlegung erzeugten Unterblöcke werden mit den Ziffern 1 bis 4 durchnummeriert. Diese Ziffern lassen sich alternativ durch die Kombination zweier Bits als Binärzahl darstellen $(00, 01, 10, 11)$ und so sehr platzsparend speichern. Die Reihenfolge der Nummerierung der Unterblöcke steht fest (siehe nachfolgende Abbildung).

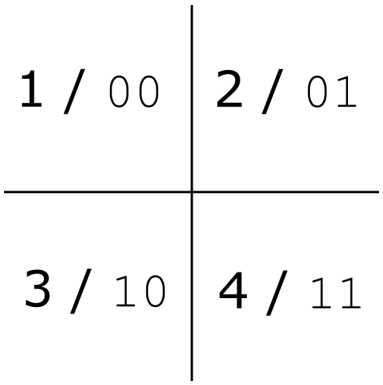


Abbildung 39: Koordinaten für Unterblöcke (Ganzzahl / Binärwert)

Die **Morton-Koordinate** eines Unterblockes ergibt sich, in man an die Morton-Koordinate des übergeordneten Blockes die Ziffer des Quadranten für den Unterblock anhängt. Die initiale Zerlegung erzeugt die vier Blöcke mit den Koordinaten $(1, 2, 3, 4)$. Wird Block 2 zerlegt, entstehen die 4 Unterblöcke $(21, 22, 23, 24)$ usw. Die maximale Anzahl der Stellen der Koordinaten ergibt, wie viele Zerlegungsstufen erfolgt sind. Für Blöcke mit kürzeren Koordinaten, welche also weniger als die maximale Anzahl zerlegt wurden, können je nach Implementierung/Anwendung bis zur maximalen Anzahl mit Nullen aufgefüllt werden. Bei drei Zerlegungstufen wird so aus der der Koordinate 21 die Koordinate 210. Dies hat den Vorteil, dass alle Ziffernfolgen gleich lang sind und sich so sehr leicht abspeichern lassen.

Das Quadtree-Konzept eignet sich sehr gut, um Rasterdaten mit großflächig homogenen Regionen (z.B. Flächenobjekten) sehr platzsparend abzubilden. Im Folgenden ist die Zerlegung beispielhaft an einem binären Raster dargestellt.

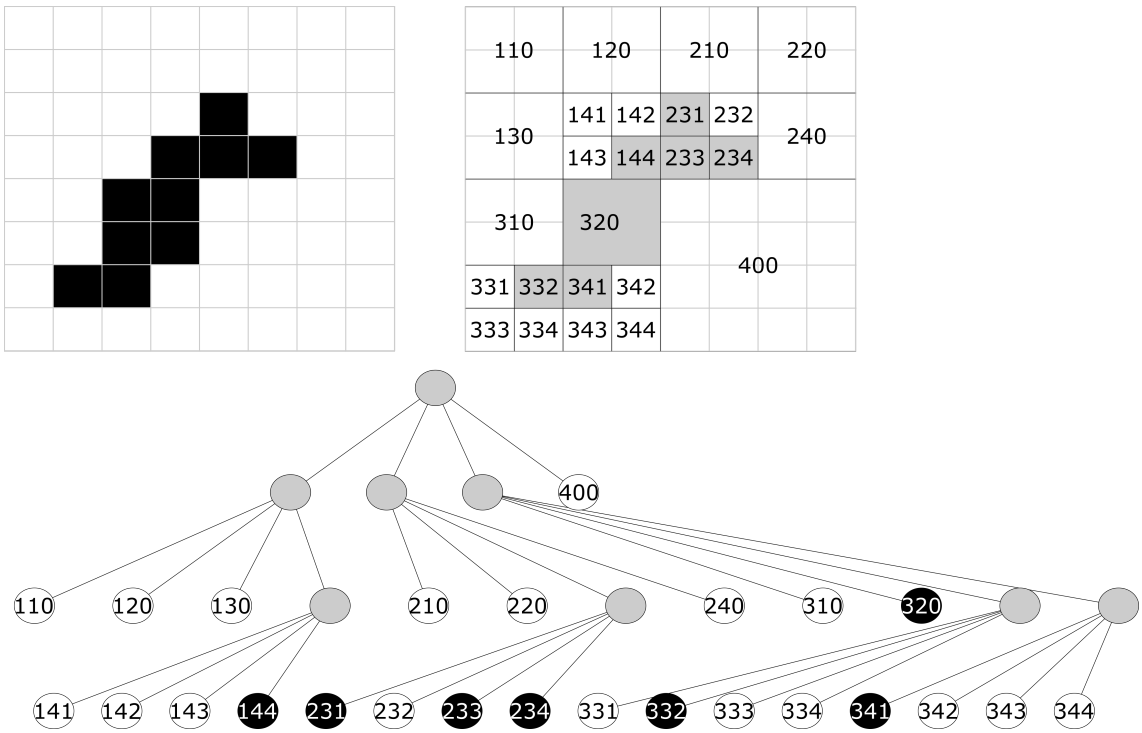


Abbildung 40: Beispielhafte Quadtree-Zerlegung eines 8 x 8 Pixel Rasters. Die Ergebnisblöcke (mit Morton-Koordinaten) enthalten entweder nur weiße oder nur schwarze Pixel (oben). Graue Knoten im Graphen (unten) repräsentieren Blöcke, die noch sowohl weiße als auch schwarze Pixel enthalten und weiter zerlegt werden müssen

Die homogenen Blöcke aus dem oben gezeigten Beispiel lassen auch über die folgende Tabelle darstellen. Die maximale Anzahl der Zerlgungsstufen muss nicht explizit vorgehalten werden, sondern ergibt sich durch die Anzahl der Stellen in den Morton-Koordinaten. Anstelle von 64 Pixelwerten beinhaltet die Tabelle nur noch 25 Einträge für Morton-Koordinaten und Attributwerte, welche dennoch das Raster vollständig repräsentieren. Die Morton-Koordinaten lassen sich, ähnlich zu den Lauf-Längen bei der Lauf-Längen-Kodierung, sehr viel effizienter abspeichern (2 Bit pro Zerlegungsstufe) als die Attributwerte selbst.

Tabelle 10: Morton-Koordinaten der homogenen Blöcke aus dem oben gezeigten Beispiel

Knoten-ID	**Morton-Koordinate**	**Attribut**
1	400	weiß
2	110	weiß
3	120	weiß
4	130	weiß
5	210	weiß
6	220	weiß
7	240	weiß
8	310	weiß
9	320	schwarz
10	141	weiß
11	142	weiß
12	143	weiß
13	144	schwarz
14	231	schwarz
15	232	weiß
16	233	schwarz
17	234	schwarz
18	331	weiß
19	332	schwarz
20	333	weiß
21	334	weiß
22	341	schwarz
23	342	weiß
24	343	weiß
25	344	weiß

Für dreidimensionale Raster wird das Konzept der Quadtrees zu so genannten Octrees erweitert. Analog zum zweidimensionalen Raster wird bei einer Octree-Zerlegung pro Zerlegungsstufe ein Rasterblock in 8 Unterblöcke zerlegt. Die zugehörigen Morton-Koordinaten verwenden jetzt nicht mehr Ziffern von 1 bis 4, sondern von 1 bis 8 und lassen sich mit max. 3 Bit pro Zerlegungsstufe $(000, 001, 010, 011, 100, 101, 110, 111)$ abspeichern.

4.3.2 Datenstrukturen für Vektordaten

Spagetti-Struktur

Eine sehr einfache Datenstruktur zur Repräsentation eines Sachverhalts im Vektormodell ist die so genannte **Spagetti-Struktur**. Die einzelnen Vektorobjekte (Punkte, Linien, Polygone) werden dabei direkt über die Listen der verwendeten Koordinaten abgebildet. So besteht zum Beispiel eine 2D-Linie aus einer Liste aufeinander folgender x,y-Koordinaten. Jede dieser Koordinaten beschreibt die Startlokation des nächsten Liniensegments. Analog dazu erfolgt Repräsentation der Grenzlinien von Polygonen. Zusätzlich können Polygone nur einfach repräsentiert werden. Dies bedeutet, dass jedes Polygon nur eine Grenzlinie aufweisen darf. Polygone mit Löchern", also mehreren inneren Grenzen, sind nicht möglich. Durch das direkte Vorhalten der Koordinatenlisten für alle Objekte lassen sich Sachverhalte in einer Spagetti-Struktur sehr effizient darstellen, da die darzustellenden Koordinaten nicht erst in anderen Tabellen oder Datenstrukturen gesucht werden müssen.

In der nachfolgenden Abbildung sind beispielhafte Tabellen für Punkt-, Linien- und Polygon-Objekte in einer Spagetti-Struktur dargestellt.

A. Point table. X and Y are locational coordinates, A_1, A_2, ...A_n are thematic attributes. Each record or row is a single point object, such as a mineral deposit location, or geochemical sample site.

ID #	X	Y	A_1	A_2	..	A_n
1	x_1	y_1	a_{11}	a_{12}	.	a_{1n}
2	x_2	y_2	a_{21}	a_{22}	.	a_{2n}
3	x_3	y_3	a_{31}	a_{32}	.	a_{3n}
.	.	.	.	.	.	.
...	.	.	.	.	.	.
m	x_m	y_m	a_{m1}	a_{m2}	.	a_{mn}

B. Line table[1]. Many lines are held in the same table or file. Each new line begins with a header (one or more records), followed by the locational coordinates of the vertices or points defining the line. In this case the first field of the header record is the line ID#, the second field is the number of vertices, and the third and fourth (or more) fields are attributes, such as feature codes. There are *m* lines.

1	5	2	7	Header for line 1
	x_1	y_1		
	x_2	y_2		Coordinates of vertices for line 1
	x_3	y_3		
	x_4	y_4		
	x_5	y_5		
2	2	4	7	Header for line 2
	x_1	y_1		Coordinates for line 2
	x_2	y_2		
3	15	2	8	Header for line 3
.	.	.		.
m	etc	etc		etc

[1] The table is nonstandard, because it contains more than one kind of record.

C. Polygon table[1]. This is essentially the same as for lines, except that the last vertex has the same coordinates as the first vertex in each polygon. Therefore there must be a minimum of four vertices per polygon. Each polygon may have many attributes, in which case the attribute data are held in a separate table, linked by polygon number. One attribute must define priority for plotting, to take care of the presence of islands. There are *m* polygons.

1	5	429	18	Header for poly 1
	x_1	y_1		
	x_2	y_2		Coordinates of vertices for polygon 1
	x_3	y_3		
	x_4	y_4		
	x_5	y_5		
2	4	39	12	Header for poly 2
	x_1	y_1		
	x_2	y_2		Coordinates for polygon 2
	x_3	y_3		
	x_4	y_4		
3	81	9	3	Header for polygon 3
.	.	.		.
m	etc	etc		etc

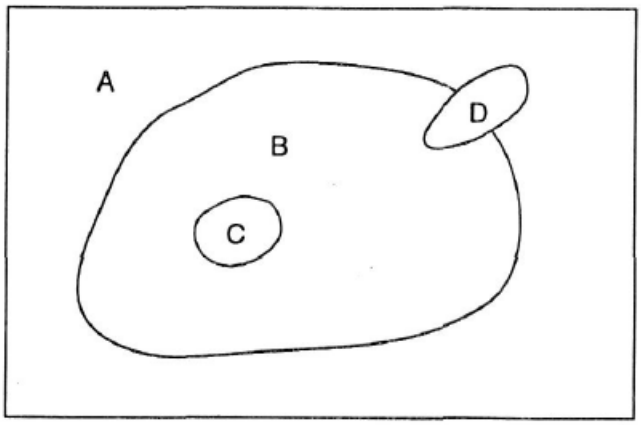
[1] This table is also nonstandard, because it contains more than one kind of record.

Table 3-7. Tables showing a very simple data structure ("spaghetti structure") for points, lines and closed polygons. Note that the spatial and non-spatial attributes are held in the same tables, and that no topological data accompanies the lines and polygons. Where polygons form an interlocking mosaic, all interior boundaries are defined twice, being part of two adjacent polygons.

Abbildung 41: Beispieltabellen für einen Vektorsachverhalt repräsentiert über eine Spagetti-Struktur ([BC94])

Die Spagetti-Struktur verwendet keine topologischen Informationen zu den repräsentierten Objekten. Dadurch werden die Beziehungen zwischen den räumlichen Objekten nicht explizit vorgehalten, sondern müssen aus den bekannten Geometrieinformationen berechnet werden. Alle Objekte werden so als voneinander unabhängig betrachtet. Mehrfach verwendete Lokationen liegen zudem mehrfach vor, jeweils einmal für jedes sie verwendende Objekt. Verwenden zum Beispiel zwei Polygone die gleichen Punktlokationen als Teil ihrer Grenze, so werden die zugehörigen Koordinaten für jedes Polygon vorgehalten. Die dadurch entstehenden Redundanzen erhöhen den Speicheraufwand und erschweren z. B. räumliche Abfragen. Um zum Beispiel zu ermitteln, ob eine beliebige Punktlokation Teil der Grenze eines oder mehrerer Polygone ist, muss sie mit allen Koordinaten aller Polygone auf Gleichheit überprüft werden. Dies ist, vor allem bei vielen Objekten, aufwändig und fehleranfällig. Aufgrund von numerischen Ungenauigkeiten und Rundungsfehlern kann es passieren, dass zwei Koordinaten zwar theoretisch gleich sein sollen, aber gespeichert eben nicht exakt gleich sind. Eine Überprüfung auf Gleichheit würde hier fehlschlagen.

Da alle Objekte unabhängig von einander vorgehalten werden, lassen sich bestimmt Sachverhalte nur umständlich realisieren. In der nachfolgenden Abbildung ist eine solche Situation dargestellt. Die Polygone A und B weisen so genannte Inselpolygone (C und D) auf. Soll diese Situation visualisiert werden, muss sichergestellt sein, das die die Polygone in der korrekten Reihenfolge darstellt werden, damit die Inselpolygone nicht verdeckt werden. Zuerst muss das größte Polygon (A) dargestellt werde, dann das Zweitgrößte (B) und erst danach die Polygone C und D. Dies kann über eine so genannte Priorisierungstabelle realisiert werden. Die Polygon werden nach aufsteigender Priorität dargestellt, erst Polygone mit niedrigen Prioritäten, dann Polygone mit höheren Prioritäten. Durch diese Polygonüberlagerung lassen sich komplexe Polygonsituationen abbilden. Weist ein Polygon ein "Loch" auf, wird dieses "Loch" als separates Polygon mit höherer Priorität vorgehalten und dargestellt. In der nachfolgenden Abbildung würde eine solche Situation z. B. auf die Polygone B und C zutreffen.



Rock	Polygon	Priority
shale	A	0
sandstone	B	1
granite	C	2
granite	D	2

FIG. 3-9. Using priority in a spaghetti structure to denote islands. Plotting sequence is determined by priority.

Abbildung 42: Polygon-Überlagerung in der Spagetti-Struktur durch Priorisierung ([BC94])

Topologie und topologische Datenstrukturen

Im Allgemeinen befasst sich die **mathematische Topologie** mit der Beschreibung von *nicht-metrischen*, *räumlichen* und *strukturellen Beziehungen* zwischen beliebigen Elementen in einem abstrakten (mathematischen) Raum ([Bil10]). Nicht-metrisch bedeutet hier, dass die Distanzen zwischen Elementen und die absolute Lage oder Form für eine topologische Betrachtung nicht relevant sind. Es wird einerseits die relative Lage räumlich verteilter Elemente zueinander betrachtet und andererseits die Beziehungen und Struktur der Elemente, das heißt welche Elemente aneinander grenzen oder wie die Elemente aus anderen Elementen aufgebaut.

Im Kontext von GIS bedeutet Topologie, dass ein realer Sachverhalt durch eine Menge von Elementen und die Beziehungen zwischen diesen Elementen beschrieben wird. Dabei wird, abhängig von ihrer Dimensionalität, zwischen verschiedenen Grund-Elementtypen unterschieden:

Punkt: Ein Punkt hat die (topologische) Dimension 0 und repräsentiert eine isolierte Position im Raum. Er hat kein Ausdehnung und ist nicht begrenzt, kann aber Teil der Grenze eines anderen, höher-dimensionalen Elements sein.

Kante: Eine Kante hat die Dimension 1. Sie verbindet immer zwei Punkte miteinander. Der Anfangs- und Endpunkt sind die Grenzen der Kante.

Fläche: Eine Fläche hat die Dimension 2. Sie ist begrenzt durch eine oder mehrere Kanten. Diese Grenzkanten bilden einen geschlossenen Ring um die Fläche.

Volumen: Ein Volumen hat die Dimension 3. Es wird durch ein oder mehrere Flächen begrenzt.

Im Kontext dieser Lehrveranstaltung sind diese Grund-Elemente (bis auf den Punkt) immer begrenzt, d. h. ein Element der Dimension d besitzt immer eine Grenze aus einem oder mehreren Elemente der Dimension $(d-1)$. Aus diesen Grund-Elementtypen lassen sich weitere wichtige topologische Elemente ableiten:

Knoten (*node*): Punkt, an dem eine Linie beginnt, endet oder an dem mehrere Linien aufeinander treffen.

Linie (*line / arc*): Zusammenhängende geordnete Menge von Kanten. Eine Linie wird durch zwei Knoten begrenzt. Der Erste entspricht dem Startpunkt der ersten Kante, der Zweite dem Endpunkt der letzten Kante. Eine Linie wird auch als Bogen (*arc*) bezeichnet.

Kette (*chain / arc*): Linie, welche Teil der Grenze eines Flächenelements (z.B. Polygon) ist. Bei einer Kette handelt es sich **immer** um eine Linie, d.h. eine Kette wird immer durch

Knoten begrenzt und umfasst eine Menge an Kanten. Eine Kette wird ebenfalls als Bogen (arc) bezeichnet. Zwei benachbarte Polygone grenzen immer entlang gemeinsamer Ketten aneinander.

Ring: Grenze eines Flächenelements (z.B. Polygon), bestehend aus einer oder mehrerer Ketten.

Polygon: Flächenelement, begrenzt durch einen oder mehrere Ringe.

einfach (*simple*): Ein einfaches Polygon weist nur einen äußeren Ring auf.

komplex Ein komplexes Polygon weist neben einem äußeren Ring einen oder mehrere innere Ringe ("Löcher") auf.

Die folgenden Abbildung stellt einen topologischen Sachverhalt beispielhaft dar. Zwei aneinander grenzende einfache Polygon (F_1 und F_2) werden durch drei Ketten (L_1, L_2 und L_3) begrenzt. Die Ketten treffen an den Knoten (K_1 und K_2) aufeinander. Die Grenze von F_1 bildet der Ring aus den beiden Ketten L_1 und L_2, analog dazu wird F_2 durch einen Ring aus den beiden Ketten L_2 und L_3 begrenzt. Die Ketten bestehen aus den Kanten (edges) E_1 bis E_7 und den Punkten (vertices) V_1 bis V_6. Der Knoten K_1 entspricht dem Punkt V_2 und der Knoten K_2 dem Punkt V_5. Die beiden Polygone grenzen entlang der gemeinsamen Kette L_2 aneinander und sind somit benachbart / adjazent.

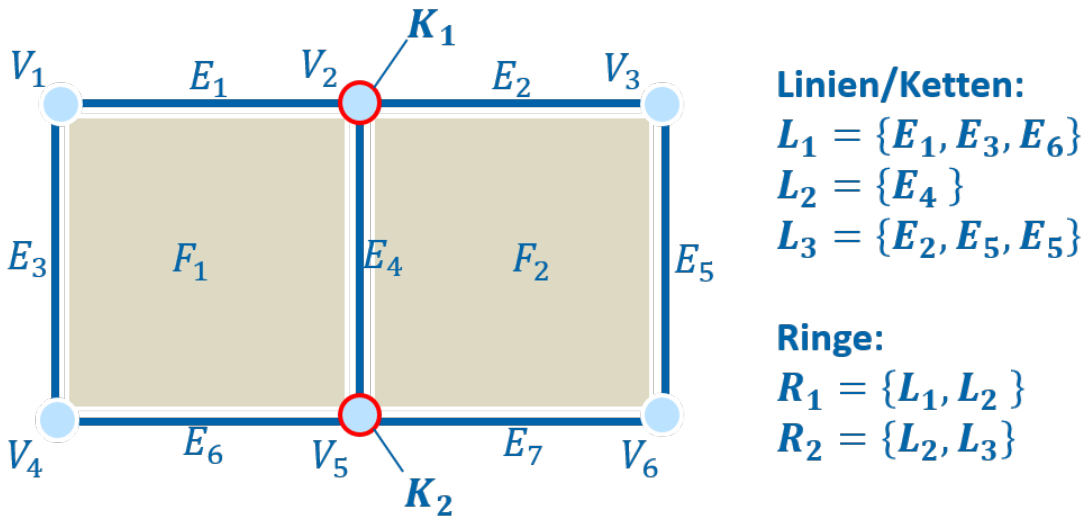


Abbildung 43: Beispiel für einen topologischen Sachverhalt mit 2 verbundenen Flächen, 3 Linien/Ketten, 7 Kanten, 2 Knoten und 6 Punkten

Die Beziehungen zwischen den topologischen Grund-Elementen werden über die Begriffe **Inzidenz** und **Adjazenz** beschrieben. Zwei Elemente sind **inzident**, wenn ein Element oder ein Teil seiner Grenze Teil der Grenze des anderen Elementes ist. So ist zum Beispiel eine Kante immer inzident zu den sie begrenzenden Punkten. Gleichzeitig sind Punkte immer inzident zu den Kanten, deren Start- oder Endpunkt sie sind. Alle Kanten, welche Teil der Grenze eines Polygons sind, sind zu diesem inzident und umgekehrt.

Zwei Elemente sind **adjazent** ("benachbart"), wenn sie einerseits inzident zueinander sind und andererseits die gleiche topologisch Dimension besitzen. Im Allgemeinen kann man auch sagen: Zwei topologische Elemente sind benachbart, wenn sie die gleiche topologische Dimension haben und ein Teil ihrer Grenze übereinstimmt. So können Flächenelemente nur adjazent zu Flächenobjekten sein, Kanten nur adjazent zu Kanten und Punkte nur adjazent zu Punkten. Zwei Punkte sind adjazent, wenn sie durch eine Kante verbunden sind. Zwei Kanten sind adjazent, wenn beide einen gleichen Punkt als Start- oder Endpunkt verwenden. Zwei Polygone sind adjazent, wenn ein Teil ihrer Grenzlinie übereinstimmt.

Im oben gezeigten Beispiel sind die beiden Flächen F_1 und F_2 inzident zueinander, da ein Teil der Grenze (Kante E_4) von F_1 ebenfalls ein Teil der Grenze von F_2 ist. Beide Fläche haben zudem

die gleiche topologische Dimension (Dimension = 2). Sie sind aus diesem Grund adjazent oder "benachbart". Die beiden Kanten E_3 und E_1 sind ebenfalls inzident, da der Punkt V_1 Teil der Grenze beider Kanten ist, und besitzen die gleiche Dimension (Dimension = 1). So sind beide Kanten ebenfalls adjazent.

Inzidenz- und Adjazenz-Beziehungen lassen sich mathematisch durch Matrizen ausdrücken. Die **Inzidenzmatrix** B betrachtet die Elemente zweier Elementtypen unterschiedlicher Dimension. Für jedes Element eines Typs wird für alle Elemente des anderen Typs angegeben, ob eine Inzidenzbeziehung besteht. B weist genauso viele Zeilen auf, wie Elemente des ersten Typs im Sachverhalt existieren und so viele Spalten, wie Elemente des zweiten Typs vorhanden sind. Für den Matrixeintrag in der i-ten Zeile und der j-ten Spalte gilt:

$$B_{ij} = \begin{cases} 1 & \text{wenn das } i\text{-te Element des ersten Typs inzident zum } j\text{-ten Element des zweiten Typs ist} \\ 0 & \text{wenn keine Inzidenz vorliegt} \end{cases}.$$

Die Inzidenzmatrix zwischen den Punkten und Kanten B_{VE} des oben gezeigten Beispiels sähe also wie folgt aus:

$$B_{VE} = \begin{array}{c|ccccccc} & E_1 & E_2 & E_3 & E_4 & E_5 & E_6 & E_7 \\ \hline V_1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ V_2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ V_3 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ V_4 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ V_5 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ V_6 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}.$$

Da eine Kante nur durch zwei Punkte begrenzt sein kann, enthält jede Spalte nur zwei Einträge, welche ungleich Null sind. Dagegen kann ein Punkt beliebig viele Kanten begrenzen und so können die Zeilen beliebig viele Werte ungleich Null enthalten.

Die **Adjazenzmatrix** A betrachtet die Elemente eines Elementtyps untereinander. Sie ist quadratisch und symmetrisch. Sie weist ebenso viele Spalten und Zeilen auf, wie Elemente des betrachteten Typs vorliegen. Für den Matrixeintrag in der i-ten Zeile und der j-ten Spalte gilt:

$$A_{ij} = \begin{cases} 1 & \text{wenn das } i\text{-te Element adjazent zum } j\text{-ten Element ist} \\ 0 & \text{wenn keine Adjazenz vorliegt} \end{cases}.$$

Da ein Element nicht zu sich selbst adjazent sein kann, gilt klassischerweise $A_{ii} = 0$, das heißt, alle Hauptdiagonalenelemente der Matrix sind immer Null. Die Adjazenzmatrix zwischen den Punkten A_V des oben gezeigten Beispiels sähe also wie folgt aus:

$$A_V = \begin{array}{c|cccccc} & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \\ \hline V_1 & 0 & 1 & 0 & 1 & 0 & 0 \\ V_2 & 1 & 0 & 1 & 0 & 1 & 0 \\ V_3 & 0 & 1 & 0 & 0 & 0 & 1 \\ V_4 & 1 & 0 & 0 & 0 & 1 & 0 \\ V_5 & 0 & 1 & 0 & 1 & 0 & 1 \\ V_6 & 0 & 0 & 1 & 0 & 1 & 0 \end{array}.$$

Adjazenz und Inzidenz sind **duale Beziehungen**. Sie lassen sich daher auseinander mathematisch herleiten. Ist die Inzidenzmatrix bekannt, lässt sich die zugehörige Adjazentmatrix berechnen und umgekehrt.

Die Begriffe Topologie, Inzidenz und Adjazenz werden in folgendem Video zur graphen-basierten Beschreibung von Vermaschungen nochmals erläutert (Prof. Gerhards; Lehrveranstaltung "Einführung in die Geoinformatik").

Van-Rössel-Topologie

Topologische Datenstrukturen beschreiben die Inzidenz- und Adjazenz-Beziehungen eines Sachverhalts im Vektorformat. Es gibt viele verschiedene Ansätze zu Datenstrukturen, um die topologischen Beziehungen so vorzuhalten, dass sie sich leicht und effizient verwalten und verarbeiten

lassen. Dabei wird jedoch in jedem Fall die Punktgeometrie von der Punktverwendung getrennt. Dies kann zum Beispiel über eine Tabelle geschehen, welche die Koordinaten eines Punktes einer eindeutigen Punktnummer zuordnet. In der weiteren Verwendung wird dann nur noch auf die eindeutige Punktnummer verwiesen. Dadurch können Punktkoordinaten nicht mehr redundant in der Datenstruktur vorkommen.

Eine sehr grundlegende topologische Datenstruktur ist die so genannte **Van-Rössel-Topologie** (van Rössel, 1987; [BC94]). Diese bildet die Topologie eines Sachverhalts über 6 vordefinierte Tabellen für Polygone, Ringe, Ketten, Knoten und Punkte ab. Einzelne Kanten werden hier nicht explizit vorgehalten. **Hinweis**: Die nachfolgend beschriebenen Tabellen entsprechen nicht den Anforderungen des relationalen Datenmodells an eine Relation. So ist zum Beispiel die Reihenfolge bestimmter Einträge in die Polygon-Tabelle entscheidend für die Bedeutung der Tabelle (äußere und innere Ringe). Grundsätzlich lassen sich die beschriebenen Tabellen aber auch in Tabellen umformen, welche konform zum relationalen Datenmodell sind.

Polygon Topologie Tabelle `polygon(polygon#,ring#,ring sequence)`:
Diese Tabelle ordnet einer Polygon-Nummer die Nummer eines Ring und die Laufrichtung des Rings (1 = Sequenz nach Ring Topologie Tabelle; 2 = Sequenz entgegen Ring Topologie Tabelle) zu. Liegen für ein Polygon mehrere Ringe vor, wird zuerst der Tabelleneintrag vorgenommen, der einer Polygon-Nummer die Nummer des äußeren Rings zuordnet, gefolgt von Einträgen, die dieser Polygon-Nummer die Nummern der inneren Ringe zuordnen.

Ring Topologie Tabelle `ring(ring#,chain#,chain sequence)`:
Diese Tabelle ordnet einer Ring-Nummer die Nummer einer Kette und die Laufrichtung der Kette (1 = Sequenz nach Ketten Topologie Tabelle; 2 = Sequenz entgegen Ketten Topologie Tabelle) zu. Liegen für einen Ring mehrere Ketten vor, werden diese in aufeinanderfolgenden Tabelleneinträgen einander zugeordnet.

Ketten Topologie Tabelle `chain(chain#,start node#, end node#, ... ... left polygon#, right polygon#)`:
Diese Tabelle ordnet einer Kette die Nummer ihres Start-Knoten, ihres End-Knotens un die Nummern für das linke und rechte Polygon zu. Hier tritt pro Kette nur exakt ein Tabelleneintrag auf. Die Laufrichtung der Ketten und damit die Richtungen "links" und "rechts" sind bestimmt durch Angabe des Start- und Endknotens. Eine Kette kann aus mehreren Kanten bestehen (siehe Kette-zu-Punkt Tabelle), alle Kanten einer Kanten einer Kette sind Teil der Grenze zwischen dem linken und rechten Polygon.

Knoten-zu-Punkt Tabelle `node2vertex(node#, vertex#)`:
Diese Tabelle ordnet jedem Knoten einen Punkt zu.

Kette-zu-Punkt Tabelle `chain2vertex(chain#, vertex#, vertex sequence)`:
Diese Tabelle beschreibt die Punktsequenz für jede Kette. Jeder Tabelleneintrag ordnet einer Ketten-Nummer die Nummer eines Punktes und die Nummer dieses Punktes in der Reihenfolge für die Kette zu. Einzelne Kanten werden in der Datenstruktur nicht explizit vorgehalten, sondern müssen aus der Punktsequenz in dieser Tabelle extrahiert werden.

Koordinatentabelle `coordinates(vertex#, x, y)`:
Diese Tabelle ordnet jedem Punkt die Koordinaten der zugehörigen Position zu.

Für das oben gezeigte Beispiel würden gemäß der Van-Rössel-Topologie folgenden Tabellen erstellt werden, um den Sachverhalt zu beschreiben.

Tabelle 11: Polygon Topologie Tabelle

polygon		
polygon#	**ring#**	**ring sequence**
F_1	R_1	1
F_2	R_2	1

Tabelle 12: Ring Topologie Tabelle

ring		
ring#	**chain#**	**chain sequence**
R_1	L_1	1
R_1	L_2	1
R_2	L_2	2
R_2	L_3	1

Tabelle 13: Ketten Topologie Tabelle

chain				
chain#	**start node#**	**end node#**	**left polygon#**	**right polygon#**
L_1	K_2	K_1	-	F_1
L_2	K_1	K_2	F_2	F_1
L_3	K_1	K_2	F_2	-

Tabelle 14: Knoten-zu-Punkt Tabelle

node2vertex	
node#	**vertex#**
K_1	V_2
K_2	V_5

Tabelle 15: Kette-zu-Punkt Tabelle

chain2vertex		
chain#	**vertex#**	**vertex sequence**
L_1	V_5	1
L_1	V_4	2
L_1	V_3	3
L_1	V_2	4
L_2	V_2	1
L_2	V_5	2
L_3	V_2	1
L_3	V_3	2
L_3	V_6	3
L_3	V_5	4

Tabelle 16: Koordinatentabelle

coordinates		
vertex#	**x**	**y**
V_1	x_1	y_1
V_2	x_2	y_2
V_3	x_3	y_3
V_4	x_4	y_4
V_5	x_5	y_5
V_6	x_6	y_6

Die Vorteile einer solchen Datenstruktur liegen auf der Hand. Zum ersten werden Redundanzen vermieden, vor allem die Punktkoordinaten sind eindeutig den Punkten zugeordnet. Zum anderen liegen zumindest einige Beziehungen zwischen den Elementen explizit vor und müssen nicht aufwändig berechnet werden. Um zum Beispiel zu ermitteln, welche Polygone benachbart sind, kann diese Frage direkt aus den Einträgen in der Tabelle
`chain(chain#,start node#, end node#, left polygon#, right polygon#)`
entnommen werden. Das linke und rechte Polygon jeder Kette sind automatisch adjazent, da sie die jeweilige Kette als gemeinsamen Teil ihrer Grenze verwenden.

Ein Nachteil solcher topologischer Datenstrukturen ist, dass die topologischen Beziehungen immer erst aufgebaut werden müssen. Dies ist abhängig von verwendeter Datenstruktur und Anzahl der topologischen Elementen zum Teil sehr aufwändig. Zudem lassen sich diese Datenstrukturen nur begrenzt effizient darstellen. Um eine Kante zu zeichnen oder auf dem Bildschirm darzustellen, muss explizit auf die Koordinaten der Endpunkte zugegriffen werden. Da in diesen Datenstrukturen die Punktkoordinaten getrennt von der Topologie verwaltet werden, ist dieser Zugriff nicht immer effizient möglich.

5 Vermaschungen

Bei einer Vermaschung oder Tesselierung (vom eng. tessellation) handelt es sich um einen speziellen Typ von Vektor-Objekten. Nach Mallet (2002, [Mal02]) werden unter diesem Begriff alle Verfahren zusammengefasst, welche es ermöglichen, ein beliebiges n-dimensionales Objekt in eine Menge **zusammenhängender, abgeschlossener, sich nicht überlappender, polytopaler n-dimensionaler** Zellen zu zerlegen. Auf diesen, geometrisch meist recht einfachen, Zellen lassen sich Berechnungen zumeist sehr viel einfacher durchführen, als auf den beliebigen Objekten selbst. Eine digitale visuelle Darstellung eines beliebigen Objektes ist ohne eine Vermaschung des Objektes in einfache diskrete Zellen oft nicht möglich. Jede Zelle einer solchen Vermaschung ist ein so genanntes **Polytop**. Ein Polytop ist die Verallgemeinerung eines zweidimensionalen Polygons auf beliebige Dimensionen. Mögliche Beispiele sind in der nachfolgenden Animation dargestellt.

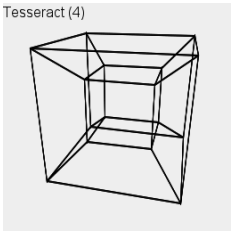


Abbildung 44: Verschiedene Polytope in verschiedenen Dimensionen (in Klammern). Quelle: https://de.wikipedia.org/wiki/Polytop_(Geometrie)

Polytope existieren für jede Dimension, am gebräuchlichsten sind Polytope der Dimensionen 0 bis 3:

Dimension 0: Punkt (Knoten)

Dimension 1: Strecke, Liniensegment

Dimension 2: Polygon, Vieleck (z.B. Dreieck)

Dimension 3: Polyeder, Vielseiter (z.B. Tetraeder, Quader)

Dimension 4: Polychor (z.B. Tesserakt)

...

Jedes Polytop der Dimension n mit $(n > 0)$ ist immer durch eine Menge von Polytopen der Dimension $(n-1)$ begrenzt. Eine Liniensegment wird so immer durch zwei Punkte begrenzt, ein Polygon durch eine Menge von Liniensegmenten, ein Polyeder durch eine Menge von Polygonen usw. Durch diese Grenze lässt sich jedes Polytop beschreiben. Dabei ist zur vollständigen Beschreibung die Kenntnis der beteiligten Punkten (0-Zellen, Knoten) immer notwendig.

Die meisten klassischen Vermaschungsverfahren zerlegen ein Objekt in konvexe Zellen/Polytope.

Konvexe Polytope und die konvexe Hülle einer Punktmenge P

Gegeben sei eine Menge von Punkten P im Raum R^n. Die konvexe Hülle $[P]$ dieser Punktmenge ist das kleinste konvexe Polytop der Dimension n, welches P komplett enthält. Dies bedeutet, dass für zwei beliebige Punkte $p \in P$ und $q \in P$, $[P]$ auch die komplette Verbindungsstrecke zwischen p und q enthalten muss. Gilt dies nicht für jedes beliebige Paar von Punkten $\in [P]$, so ist [P] nicht konvex und kann damit nicht die konvexe Hülle der Punktmenge P sein. In der nachfolgenden Abbildung ist (a) eine konvexe Menge, da für alle Punkte auch ihre Verbindungen in der Menge enthalten sind. Für (b) gilt dies nicht.

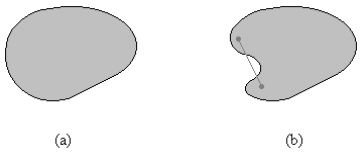


Abbildung 45: (a) konvexe Menge, (b) nicht-konvexe / konkave Menge

In der Ebene wird die konvexe Hülle einer Punktmenge durch das kleinste konvexe Polygon definiert, welches neben allen Punkten der Menge auch alle Punktverbindungsstrecken enthält. Aus diesem Grund ist das in der nachfolgenden Abbildung dargestellte Polygon (schwarz) nicht konvex. Die Verbindungsstrecke zwischen den zwei Punkten p und q (blau) ist nicht komplett enthalten. Die konvexe Hülle dieses Polygons ist als rote gestrichelte Linie dargestellt.

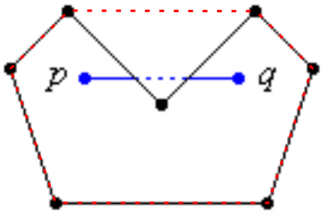


Abbildung 46: Konkaves oder nicht-konvexes Polygon (schwarz), konvexe Hülle (rot)

Simplex, Simplices und Simplizialkomplexe

Ein **Simplex** der Dimension n ist ein Polytop der Dimension n, welches immer $(n+1)$ Randelemente aufweist. Dazu zählen unter anderen das Liniensegment (Dim. 1), das Dreieck (Dim. 2) oder der Tetraeder (Dim. 3). Ein Simplex ist **immer** ein konvexes Polytop. Eine Vermaschung, welche nur aus Simplices einer Dimension besteht, wird als **Simplizialkomplex** bezeichnet. Dazu zählen zum Beispiel beliebige Zerlegung eines Objektes in Dreiecke oder Tetraeder, welche auch als **Triangulierungen** bezeichnet werden. Auch ein Linienobjekt, bestehend aus einer Menge aufeinander folgender Liniensegmente, ist ein Simplizialkomplex.

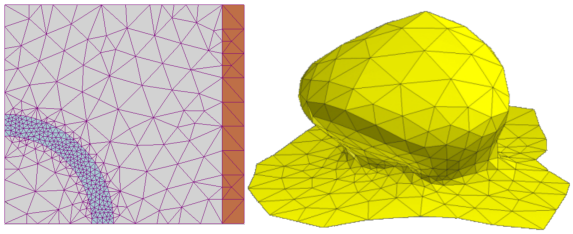


Abbildung 47: Simplizialkomplexe der Dimension 2: Zerlegung von Flächenobjekten in Dreiecke

Euler-Poincaré-Theorem: Für jede beliebige Zerlegung / Vermaschung $\mathbf{P}(A)$ eines n-dimensionalen Objekts A lässt sich die Euler-Poincaré-Charakteristik $\chi(A)$ wie folgt berechnen ([Mal02]):

$$\chi(A) = \sum_{i=0}^{n} (-1)^i \cdot |C_i(A)|.$$

Dabei ist $|C_i(A)|$ die Anzahl aller Zellen mit topologischer Dimension i in $\mathbf{P}(A)$. Für eine zweidimensionale Fläche gilt demzufolge $\chi = V - E - F = 2 - n_{Grenzen}$. V ist die Anzahl der Knoten / Vertices, E die Anzahl der Kanten und F die Anzahl der Flächenelemente (Polygone). $n_{Grenzen}$ gibt an, wie viele Grenzen das Flächenobjekt A aufweist. Beispiele dafür sind in der folgenden Abbildung dargestellt.

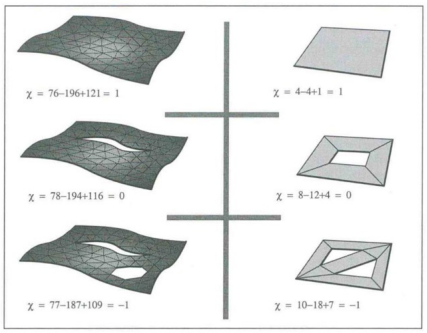


Figure 2.12 *Examples of cellular partitions of open surfaces: the Euler-Poincaré characteristic $\chi = V - E + F$ is equal to 2 minus the number of boundaries whatever the cellular partition.*

Abbildung 48: Beispiele für das Euler-Poincaré-Theorem ([Mal02])

5.1 Voronoi-Vermaschung

Eine **Voronoi-Vermaschung** (auch Voronoi-Zerlegung oder Voronoi-Diagramm) erlaubt die Zerlegung eines Gebietes Ω in eine Menge konvexer polytopaler Zellen basierend auf einer gegebenen Menge an Punkten P innerhalb des Gebietes. Eine solche Vermaschung wird alternativ auch als **Thiessen-Vermaschung / -Zerlegung** oder **Dirichlet-Vermaschung / -Zerlegung bezeichnet**. Die verschiedenen Bezeichnung rühren daher, dass diese Art von Vermaschungen zu verschiedenen Zeiten von verschiedenen Personen unabhängig voneinander beschrieben wurden (erstmals beschrieben von Dirichlet im Jahre 1850).

Die Ausgangssituation für die Vermaschung ist eine Punktmenge P aus m verschiedenen Punkten $P = \{p_1, \ldots, p_m\}$ in einem n-dimensionalen euklidischen Raum R^n. Das Voronoi-Polytop $V(p_i)$ für einen Punkt $p_i \in P$ ist ein Unterraum von R^n, der alle Punkte x beinhaltet, die "näher" an dem Punkt p_i liegen als an jedem anderen Punkt aus P mit

$$V(p_i) = \{x \in R^n | d(p_i, x) < d(p_j, x) \forall j \neq i\}.$$

$d(p_i, x)$ ist dabei der euklidische Abstand zwischen p_i und x.

Ein Beispiel für eine Voronoi-Vermaschung ist in der folgenden Abbildung dargestellt. Die Vermaschung basiert auf den roten Punkten (Zellzentren), die Voronoi-Zellen werden durch die blauen Linien begrenzt. Jeder beliebige Punkt innerhalb einer solchen Zelle weist zum Zentrum der Zelle einen geringeren Abstand auf, als zu allen anderen Zellzentren. Das Zentrum einer Zelle $V(p_i)$ ist immer der Punk p_i. Ein Punkt auf der Grenze zwischen zwei Voronoi-Zellen (blaue Linien) weist zu den beiden Zentren der angrenzenden Zellen den gleichen Abstand auf, d.h. ist gleich weit von beiden Zentren entfernt.

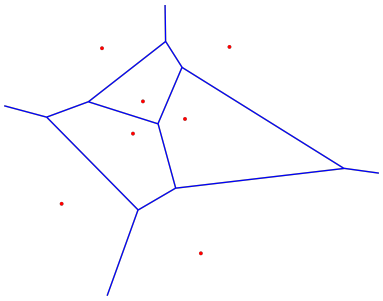


Abbildung 49: Beispiel für eine 2D Voronoi-Vermaschung basierend auf den roten Punkten. Die blauen Linien trennen die verschiedenen Zellen (Quelle: https://de.wikipedia.org/wiki/Voronoi-Diagramm)

Die Voronoi-Vermaschung $\mathbf{V}(P)$ von P umfasst alle Voronoi-Zellen zu allen Punkten aus P mit $\mathbf{V}(P) = \{V(p_i), 1 \leq i \leq m\}$. Zwei Voronoi-Zellen und damit auch die ihnen zugrunde liegenden Punkte werden als **kanonische** (*canonical*) oder **natürliche** (*natural*) Nachbar bezeichnet, wenn

beide Zellen direkt aneinander grenzen und so einen Teil ihrer Grenze gemeinsam haben. So sind in der nachfolgenden Abbildung die beiden Zellen $V(p_1)$ und $V(p_2)$ natürliche Nachbarn. Für $V(p_1)$ und $V(p_4)$ trifft dies nicht zu, weil sie keine gemeinsame Grenzlinie haben. An so genannten **Knoten** der Voronoi-Vermaschung treffen die Grenzlinien mehrerer Voronoi-Zellen aufeinander. Der Punkt q in der folgenden Abbildung ist zum Beispiel ein solcher Knoten.

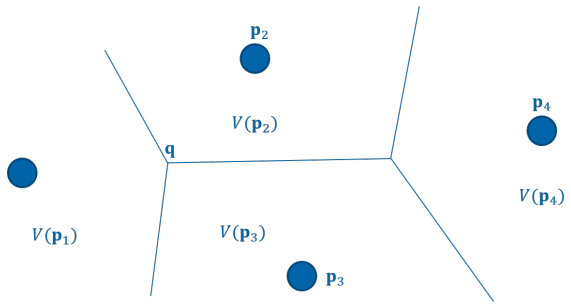


Abbildung 50: Voronoi-Vermaschung $\mathbf{V} = \{V(p_1), \ldots, V(p_4)\}$ basierend auf den 4 Punkten p_1 bis p_4. Am Knoten q treffen 3 Zellen aufeinander

Eine Voronoi-Vermaschung $\mathbf{V}(P)$ hat folgenden Eigenschaften:

1. Jede Voronoi-Zelle $V(p_i)$ mit $p_i \in P$ ist **konvex**.
2. Das Zellzentrum p_i der Zelle $V(p_i)$ befindet sich **immer** innerhalb der Zelle.
3. Die Anzahl der natürlichen Nachbarn einer Voronoi-Zelle und die Anzahl der Knoten auf der Grenze einer Voronoi-Zelle steht nicht fest und kann für jede Zelle einer solchen Vermaschung verschieden sein.
4. Befindet sich ein Punkt $p_i \in P$ auf der **konvexen Hülle** $[P]$ von P, so ist die zugehörige Voronoi-Zelle $V(p_i)$ nicht vollständig begrenzt, d.h. im 2D-Fall: ihre Grenzlinie bildet keinen geschlossenen Ring. Die Zelle $V(p_i)$ umfasst daher einen unendlichen großen Unterraum von R^n.
5. Für eine Voronoi-Vermaschung im R^n treffen in einem Knoten q im Allgemeinen die Grenzen von $(n+1)$ Voronoi-Zellen aufeinander. In einem Kreis mit diesem Knoten als Zentrum, der durch die Zellzentren der angrenzenden Zellen verläuft, liegen **keine** anderen Zell-Zentren.

Die Konstruktion der exakten Grenzen einer Voronoi-Zelle ist vergleichsweise aufwändig. Ausgehend von zwei Punkten p_i und p_j im Raum R^n lässt sich dieser Raum in zwei Unterräume $H(p_i, p_j)$ und $H(p_j, p_i)$ aufspalten. Für den Unterraum $H(p_i, p_j)$ gilt Folgendes:

$$H(p_i, p_j) = \{x \in R^n | d(p_i, x) < d(p_j, x)\},$$

das heißt, dass der Unterraum $H(p_i, p_j)$ alle Positionen x umfasst, welche "näher" am Punkt p_i als am Punkt p_j liegen. Der Unterraum $H(p_j, p_i)$ umfasst analog dazu alle Positionen x, welche "näher" am Punkt p_j als am Punkt p_i liegen.

Beide Unterräume werden durch den so genannten Bisektor $B(p_i, p_j)$ getrennt, für den gilt

$$B(p_i, p_j) = \{x \in R^n | d(p_i, x) = d(p_j, x)\}.$$

Es handelt sich um eine Hyperebene im Raum R^n (zum Beispiel eine Linie im R^2 oder eine Ebene im R^3) auf der eine beliebige Position $x \in B(p_i, p_j)$ von p_i und p_j gleich weit entfernt ist. Es gilt also $d(p_i, x) = d(p_j, x)$. In der folgenden Abbildung ist dies schematisch für eine Situation in 2D dargestellt. Der Punkt x_1 liegt auf dem Bisektor (gestrichelte Linie), für ihn gilt somit

$$x_1 \in B(p_i, p_j) \rightarrow d(x_1, p_i) = d(x_1, p_j).$$

Der Punkt x_2 liegt im Halbraum $H(p_i, p_j)$, für ihn gilt somit

$$x_2 \in H(p_i, p_j) \rightarrow d(x_2, p_i) < d(x_2, p_j).$$

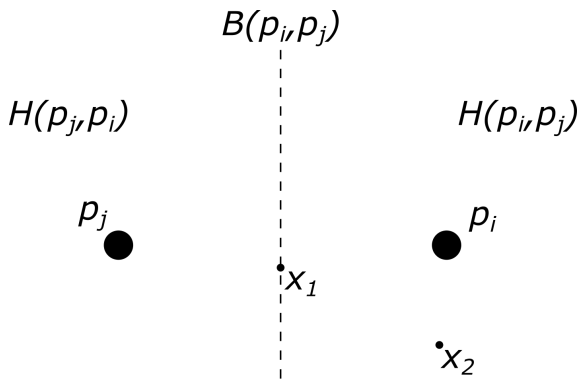


Abbildung 51: Schematische Darstellungen der Halbräume $H(p_i, p_j)$ und $H(p_j, p_i)$ und des Bisektors $B(p_i, p_j)$

Mathematisch ergibt sich das Innere der Voronoi-Zelle $V(p_i)$ als die Schnittmenge aller Halbräume bezüglich p_i und allen anderen Punkten $p_j \in P, j \neq i$ mit $V(p_i) = \bigcap_{j=1;j\neq i}^{m} H(p_i, p_j)$.

In der nachfolgenden Animation wird demonstriert, wie die Erstellung einer Voronoi-Zelle in 2D praktisch ablaufen kann. Für den magenta-farbenen Punkt wird die Grenze der zugehörigen Zelle erzeugt, indem sukzessive die Bisektoren zu allen anderen Punkten konstruiert werden (blaue und rote Linien). Schneiden sich zwei Bisektoren, wird der Schnittpunkt als Knoten (rote Punkte) und das entstehende Liniensegment der Zellgrenze hinzugefügt, wenn sich zwischen dem Schnittpunkt / Liniensegment und dem Zellzentrum keine vorherigen Grenzsegmente befinden. Dabei kann es passieren, dass ein vorher erstellter Knoten durch einen oder mehrere neue Knoten ersetzt wird (rote Kreuze). Die gestrichelten Bisektoren geben an, dass in diesem Fall keine neue Knoten / Segmente erstellt wurden. Einmal feststehende Knoten und Grenzsegmente werden in den Graph der Vermaschung übernommen. Dieses Vorgehen muss für alle Zellzentren wiederholt werden, um die komplette Vermaschung zu erhalten.

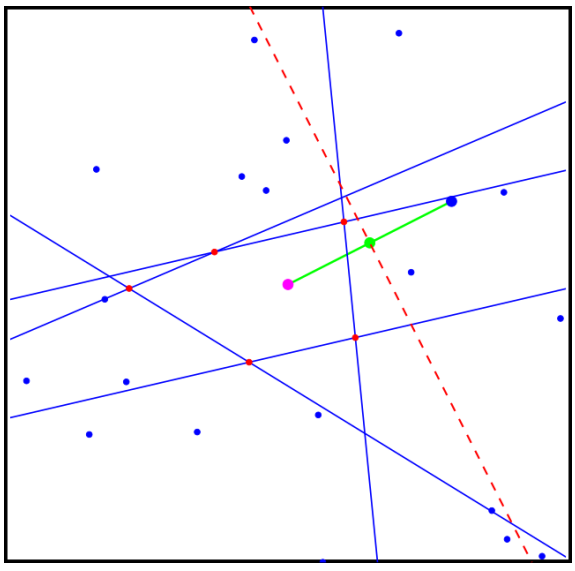


Abbildung 52: Animation zur beispielhaften Erzeugung einer Voronoi-Zelle zum magenta-farbenen Punkt

Die Voronoi-Vermaschung verknüpft das Zentrum einer Zelle mit allen Knoten der Voronoi-Zelle und den Zellgrenzen, welche aus den beteiligten Segmenten der Bisektoren bestehen. Eine allgemeine Datenstruktur zur Repräsentation einer Voronoi-Vermaschung in 2D besteht aus einer Liste / Tabelle `nodes(nodeID,x,y)`, welche allen Knoten einen Identifizierer und die zu-

gehörigen Koordinaten zuordnet und einer Liste / Tabelle `voronoiCells(cellID,nodeList)`, welche einer Zelle die geordnete Liste der beteiligten Knoten zuordnet. Diese Liste der beteiligten Knoten muss es erlauben, für jede Zelle auf eine beliebige Anzahl von Knoten zu verweisen, da die Anzahl der Knoten einer Zelle nicht feststeht. Zusätzlich können über eine weitere Liste `cellCenters(cellID, x_center, y_center)` jeder Zelle die Koordinaten des Zellmittelpunktes zugeordnet werden. Die benötigten Tabellen könnten die folgenden Strukturen aufweisen:

Tabelle 17: nodes

nodes		
nodeID	**x**	**y**
1	x_1	y_1
⋮	⋮	⋮
max. Anzahl der Nodes n	x_n	y_n

Tabelle 18: voronoiCells

voronoiCells	
cellID	**nodeList**
1	nodeID_1, nodeID_2, nodeID_3, ...
⋮	⋮
max. Anzahl der Zellen m	nodeID_i, nodeID_j, nodeID_k, ...

Tabelle 19: cellCenters

cellCenters		
cellID	**x_center**	**y_center**
1	x_1	y_1
⋮	⋮	⋮
max. Anzahl der zellen m	x_m	y_m

Für Voronoi-Vermaschungen in 3D oder höheren Dimensionen ist eine sehr viel kompliziertere Datenstruktur notwendig, da sich beliebige n-Zellen nicht mehr über eine einfache geordnete Liste der Grenzknoten beschreiben lassen. In 3D müssen mindestens für jede 3-Zelle alle Grenzpolygone über ihre Grenzlinien repräsentiert werden. Dafür könnten zum Beispiel kombinatorische Datenstrukturen wie Generalized Maps ([Mal02]) verwendet werden.

Eine weitere Einführung zu Voronoi-Vermaschungen finden Sie in einem ScreenCast aus der Lehrveranstaltung "Einführung in die Geoinformatik".

5.2 Triangulierungen

Grundsätzlich ist jede zweidimensionale Zerlegung der konvexen Hülle $[P]$ einer Punktmenge P in Dreiecke eine **Triangulierung**. P besteht aus m eindeutigen, nicht-kollinearen Punkten $p_i = p(x_i, y_i), i = 1, \ldots, m$. Eine ebene Triangulierung $\mathbf{T}(P)$ besteht aus einer Menge an Punkt-Tripeln (p_i, p_j, p_k) (Dreiecken), für die gilt:

- Jedes Tripel / Dreieck ist **nicht-degeneriert**. Dies bedeutet, dass alle drei Punkte verschieden sind mit $i \neq j \neq k$.
- Ein Dreieck ist **immer komplett begrenzt** von drei Liniensegmenten zwischen den drei zugehörigen Punkten.
- **Kein Dreieck** enthält einen oder mehrere weitere Punkte $p_l \in P$ mit $l \neq i, j, k$.
- Die Schnittmenge zweier Dreiecke ist **leer** mit $(p_i, p_j, p_k) \cap (p_l, p_m, p_n) = \emptyset$. Die Dreiecke überlappen sich nicht.
- Die Vereinigung aller Dreiecke bildet die **konvexe Hülle** $[P]$ von P.

In der folgenden Abbildung ist eine mögliche Triangulierung der gegebenen Punkte (schwarz) dargestellt. Die konvexe Hülle der Punkte ist über die rote gestrichelte Linie dargestellt.

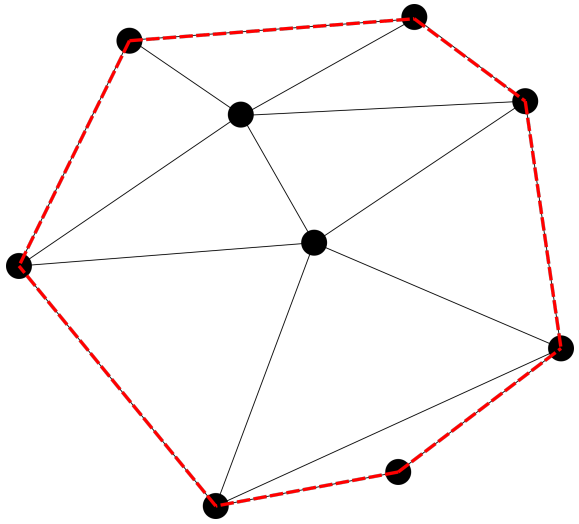


Abbildung 53: Mögliche Triangulierung der schwarzen Punkte

Für eine Punktmenge P existieren immer mehrere mögliche Triangulierungen. Basierend auf dem *Euler-Poincaré-Theorem* (siehe Abschnitt *Einführung* zum Thema *Vermaschungen*) lässt sich aber die Anzahl der Flächenelemente (Dreiecke) F über

$$F = 2m - n_{\text{konvexe Hülle}} - 2$$

und die Anzahl der Dreieckskanten E über

$$E = 3m - n_{\text{konvexe Hülle}} - 3$$

abschätzen mit $m - 1 \leq F \leq 2m - 5$ und $3m - 3 \leq E \leq 3m - 6$. $n_{\text{konvexe Hülle}}$ beschreibt die Azahl der Punkte aus P, welche sich auf der konvexen Hülle $[P]$ von P befinden, m ist die Anzahl der Punkte in P.

Triangulierungskriterien

Um eine bestimmte Triangulierung zu erhalten, werden bestimmte Modellannahmen und Auswahlkriterien benötigt. Diese werden auch als **Triangulierungskriterien** bezeichnet. Diese Kriterien beziehen sich immer darauf, wie die Triangulierung eines konvexen Vierecks erfolgen soll. Für jedes

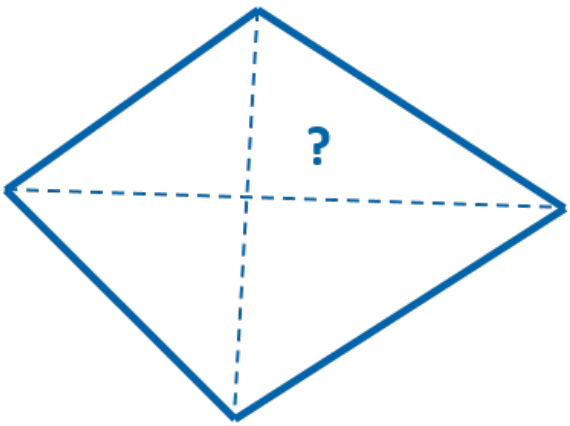


Abbildung 54: Mögliche Triangulierungen eines konvexen Vierecks

konvexe Viereck gibt es immer zwei mögliche Zerlegungen in zwei Dreiecke, wie die nachfolgenden Abbildung skizziert. Die Triangulierung erfolgt entlang **einer** der beiden gestrichelten Kanten.

Mögliche Kriterien sollen zum Beispiel erreichen, dass die erstellten Dreiecke global so gleichförmig wie möglich sind, also möglichst gleichseitig (*equi-lateral*) oder möglichst gleichwinklig (*equi-angular*). Ein anderes Kriterium könnte sein, dass Dreiecke mit extremen Winkeln vermieden werden sollen. Dafür müssten die Dreiecke dahingehend optimiert werden, dass entweder die maximalen Winkel minimiert (**Min-Max-Kriterium**) oder die minimalen Winkel maximiert (**Max-Min-Kriterium**) werden.

Beispiele für Triangulierungskriterien sind:

- **Maximierung** des **Kleinsten** der 6 Winkel von 2 Dreiecken eines konvexen Vierecks (**Max-Min** / *Delaunay Triangulierung*)
- **Minimierung** des **Größten** der 6 Winkel von 2 Dreiecken eines konvexen Vierecks (**Min-Max**)
- Triangulierung entlang der **kürzesten / längsten** Diagonale eines konvexen Vierecks

Weitere Informationen zu Triangulierungen finden Sie in verschiedenen ScreenCasts aus der Lehrveranstaltung "Einführung in die Geoinformatik".

5.2.1 Delaunay Triangulierung

Die duale Vermaschung zur Voronoi-Vermaschung

Die **duale Vermaschung** einer Voronoi-Vermaschung $\mathbf{V}(P)$ wird als **Delaunay Vermaschung** $\mathbf{D}(P)$ bezeichnet. Zwei n-dimensionale Vermaschungen $\mathbf{G}(P)$ und $\mathbf{G^*}(P)$ sind **dual zueinander**, wenn jeder i-Zelle aus $\mathbf{G}(P)$ eine $(n-i)$-Zelle aus $\mathbf{G^*}(P)$ zugeordnet wird und umgekehrt. In 2D wird so jeder Voronoi-Zelle (Dimension $i = n = 2$) aus $\mathbf{V}(P)$ ein Knoten (Dimension $i = n-2 = 0$) aus $\mathbf{D}(P)$ zugeordnet. Die Knoten der Delaunay Vermaschung liegen exakt an der Position, an der das Zellzentrum der dualen Voronoi-Zelle liegt. Jedem Voronoi-Knoten (0-Zelle) wird analog eine 2-Zelle (Dreieck) der Delaunay Vermaschung zugeordnet. Dabei entspricht der Voronoi-Knoten dem Zentrum des Umkreises der zugehörigen Delaunay-Zelle. Zumindest in 2D werden Kanten (1-Zellen) aus der Voronoi-Vermaschung auf Kanten (1-Zellen) der Delaunay Vermaschung abgebildet. Beide Kanten stehen im Allgemeinen senkrecht aufeinander.

Diese duale Beziehung ist exemplarisch in 2D in der folgenden Abbildung dargestellt. Die Knoten und Kanten der Voronoi-Vermaschung sind als weiße Punkte und Linien dargestellt, die Knoten und Kanten der zugehörigen Delaunay Vermaschung sind als schwarze Punkte und Linien. Die schwarzen Punkte sind gleichzeitig äquivalent zu den Zellzentren der Voronoi-Vermaschung.

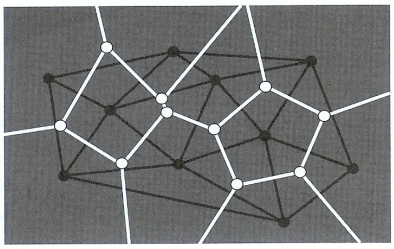


Figure 3.1 *An example of a 2-dimensional Voronoi's diagram (in white) and its associated Delaunay triangulation (in black). The Voronoi regions are represented in light grey. The Delaunay points are represented by black points, while the Voronoi points are represented by white points.*

Abbildung 55: Dualität zwischen einer Voronoi-Vermaschung (weiß) und einer Delaunay Vermaschung (schwarz) ([Mal02])

In der folgenden Animation ist die Dualität der Voronoi-Zellen $V(p_1)$ bis $V(p_4)$ (blau) zu den beiden Delaunay-Dreiecken $T(q_i)$ und $T(q_j)$ (schwarz) gezeigt. Die Voronoi-Zellzentren p_1 bis p_4 (blau) entsprechen den Knoten der Delaunay-Dreiecke.

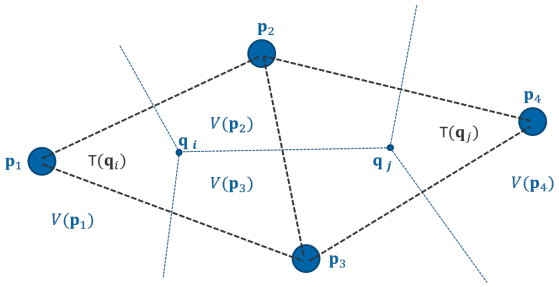


Abbildung 56: Dualität der Voronoi-Zellen $V(p_1)$ bis $V(p_4)$ zu den beiden Delaunay-Dreiecken $T(q_i)$ und $T(q_j)$

Bei der Delaunay Vermaschung handelt es sich um eine Triangulierung (Delaunay Triangulierung), also um eine Zerlegung in Dreiecke (2D) oder Tetraeder (3D). Alle n-Zellen sind simplizial, das heißt,

- alle n-Zellen werden durch $(n+1)$ $(n-1)$-Zellen begrenzt und
- alle n-Zellen weisen $(n+1)$ natürliche Nachbarn auf.

Die Vermaschung einer Menge von Punkten P im R^n, in der die natürlichen Voronoi-Nachbarn so verbunden sind, dass sich n-Simplizes (2D: Dreiecke, 3D: Tetraeder) bilden, wird als Delaunay Triangulierung bezeichnet und weißt folgenden spezielle Eigenschaften auf:

- **Die Grenzelemente (alle $(n-1)$-Zellen, die nicht an zwei n-Zellen grenzen) der Triangulierung D(P) bilden die konvexe Hülle der Punktmenge P.**
- In 2D ist die Delaunay Triangulierung die Triangulierung, **die den kleinsten Winkel jedes Dreiecks maximiert**.

Diese Eigenschaften können direkt zum Aufbau der Triangulierung verwendet werden. Die Kenntnis der zugehörigen Voronoi-Vermaschung ist dafür nicht notwendig.

Konstruktive Delaunay Triangulierung

Die Delaunay Triangulierung maximiert immer den kleinsten Winkel von zwei Dreiecken. Das Max-Min-Kriterium ist äquivalent zum so genannten **Umkreiskriterium**. Aufgrund der Dualität zur Voronoi-Vermaschung ergibt sich dieses Kriterium direkt aus der Eigenschaft einer Voronoi-Vermaschung, nach der der Kreis um einen Voronoi-Knoten durch die angrenzenden Zellzentren kein weiteres Zellzentrum enthalten darf (siehe 5. Eigenschaft im Abschnitt *Voronoi-Vermaschung*).

Der Umkreis eines Dreiecks ist genau der Kreis, auf dem die drei Knoten des Dreiecks liegen. Der Mittelpunkt OO dieses Kreises ist der Schnittpunkt der drei Mittelsenkrechten des Dreiecks. In 2D ist eine Triangulierung einer Punktmenge PP genau dann die **Delaunay Triangulierung** dieser Punkte, wenn der Umkreis jedes Dreiecks keinen weiteren Punkt aus P beinhaltet, als die drei Punkte des Dreiecks.

In der nachfolgenden Animation wird dieses Umkreiskriterium am Beispiel der zwei möglichen Triangulierungen für 4 Punkte dargestellt. Die linke Triangulierung ist die Delaunay Triangulierung. Die Umkreise (grau, gestrichelt) der beiden Dreiecke enthalten nie den vierten Punkt. Bei der rechten Triangulierung ist der vierte Punkte (rot) immer im Umkreis des Dreiecks der drei anderen Punkte enthalten. Es handelt sich gemäß des Umkreiskriteriums also nicht um die Delaunay Triangulierung dieser Punkte.

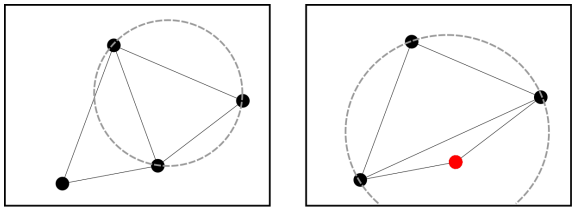


Abbildung 57: Delaunay (links) und Nicht-Delaunay (rechts) Triangulierung von vier Punkten

Lokale Optimalität einer Triangulierung Eine Triangulierung wird als **lokal optimal** bezeichnet, wenn jedes konvexe Viereck bezüglich des gewählten Triangulierungskriteriums optimal trianguliert wurde. Dies lässt sich leicht über den so genannten **Diagonalen-Tausch-Test** erreichen. Für jedes konvexe Viereck lassen sich die beiden möglichen Triangulierungen mittels des Vertauschens der Diagonalen erzeugen (gestrichelte Linien in der folgenden Abbildung). Beide Triangulierungen werden hinsichtlich des Triangulierungskriteriums verglichen und es wird die gewählt, welche dem Kriterium entspricht.

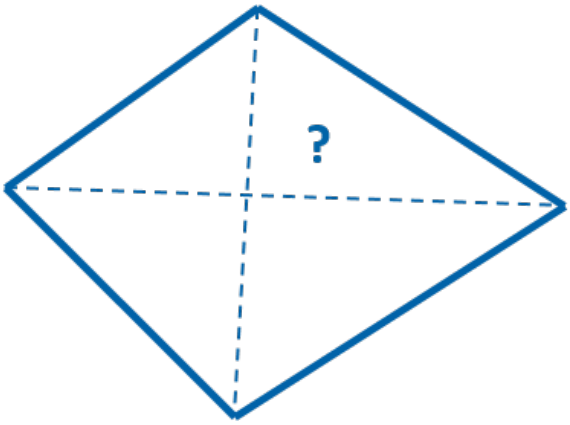


Abbildung 58: Mögliche Triangulierungen eines konvexen Vierecks

Globale Optimalität einer Triangulierung Für eine gegebene Punktmenge P können mehrere lokal optimale Triangulierungen existieren. Um zu entscheiden, welche dieser Triangulierungen auch **global optimal** ist, ist ein Maß notwendig, um die lokal optimalen Triangulierungen miteinander vergleichen zu können.

Sei $a(T), T \in \mathbf{T}$ ein Maß für die Form eines einzelnen Dreiecks. Der Vektor

$$a(\mathbf{T}) = (a_1, a_2, \ldots, a_N), a_i = a(T_i), T_i \in \mathbf{T}, i = 1, \ldots, N$$

wird einer Triangulierung $\mathbf{T}$ zugeordnet, die Komponenten a_i sind aufsteigend sortiert und beschreiben z.B. den Kleinsten der drei Winkel im Dreieck T_i.

Eine Triangulierung $\mathbf{T}_1$ wird als "besser"(engl. superior) bezüglich eines Maßes a als eine Triangulierung $\mathbf{T}_2$ bezeichnet, wenn gilt: $a(\mathbf{T}_1) > a(\mathbf{T}_2)$.

Der Operator $>$ bedeutet hier lexikografische Reihenfolge:
Für zwei Vektoren c und b bedeutet $c < b$, dass eine Zahl $k \leq N$ existiert, für die gilt:

$$c_i = b_i, i = 1, \ldots, k-1 \text{ und } c_k < b_k.$$

Eine Triangulierung $\mathbf{T}^*$ einer Punktmenge P ist **global optimal bezüglich Maß** a, wenn für alle Triangulierungen $\mathbf{T}(P)$ gilt:

$$a(\mathbf{T}^*) \geq a(\mathbf{T})$$

Dies bedeutet, dass es keinen lexikografisch größeren Vektor als $a(\mathbf{T}^*)$ gibt. Der Wert k für diesen Vektor $a(\mathbf{T}^*)$ ist größer als für alle anderen Vektoren $a(\mathbf{T})$. Im Allgemeinen gilt für $a(\mathbf{T}^*)$ sogar $k = N$.

Jede global optimale Triangulierung ist immer auch lokal optimal! Dieser Fakt gilt nicht notwendigerweise umgekehrt; nur weil eine Triangulierung lokal optimal ist, muss sie nicht global optimal sein. Im Fall des Max-Min-Kriteriums trifft dies jedoch tatsächlich zu:
Eine bezüglich des Max-Min-Kriteriums lokal optimale Triangulierung ist auch global optimal! Es ist kein anderes Kriterium bekannt, für welches dieser Fakt ebenfalls gilt.

Algorithmen zur Erstellung einer Delaunay Triangulierung Die Delaunay Triangulierung, als duale Vermaschung der Voronoi-Vermaschung, ist lokal optimal bezüglich des Max-Min-Kriteriums **UND** zugleich global optimal bezüglich des Max-Min-Kriteriums. Diese Eigenschaften lassen sich verwenden, um effiziente Triangulierungsalgorithmen zu entwerfen. Diese starten immer damit, dass sukzessiv jedes mögliche Viereck lokal optimal bezüglich des Max-Min.Kriteriums trianguliert wird. Die erhaltene Triangulierung ist dann automatisch auch global optimal. Es lassen sich drei unterschiedliche Typen von Algorithmen ableiten:

- Für jedes Viereck einer bereits gegebenen Triangulierung wird geprüft, ob das Max-Min-Kriterium erfüllt ist. Wenn nicht, wird die Diagonale vertauscht.
- Es wird mit einem möglichen Dreieck gestartet. Sukzessiv werden weitere hinzugefügt und jedes entstehende konvexe Viereck wird lokal optimal trianguliert.
- Die Punktmenge P wird in kleinere zusammenhängende Teilmengen unterteilt und diese werden lokal optimal trianguliert. Danach werden alle Teiltriangulierungen so zusammengefügt, dass die lokale Optimalität erhalten bleibt.

5.2.2 Bedingte Delaunay Triangulierung

Für bestimmte Anwendungen existieren "a priori" Informationen zu den natürlichen Nachbarschaftsbeziehungen in P. Diese können zum Beispiel sein:

- Grenzlinie einer bekannten, nicht-konvexen Grenzpolygons einer zu triangulierenden Punktmenge.
- Isolinien der Höhe oder anderer kontinuierlicher Parameterverteilungen. Die Punkte auf einer solchen Linien weisen den gleichen Parameterwert auf; adjazente Punkte auf einer Isolinie sind natürliche Nachbarn.
- Störungslinien / Diskontinuitäten; adjazente Punkte auf einer Seite der Störung sind natürliche Nachbarn, aber keine Nachbarschaft über die Störung hinweg.

Diese "a prori" Nachbarschaften äußern sich in Linienobjekten, welche durch die Triangulierung berücksichtigt werden müssen. So müssen alle Kanten des Grenzpolygons oder die die Isolinien repräsentierenden Kantenzüge in der Triangulierung erhalten bleiben. Bezüglich Störungen / Diskontinuitäten muss sichergestellt werden, dass keine durch die Triangulierung erzeugte Kante die Störung / Diskontinuitäten kreuzt. Der die Störung / Diskontinuität repräsentierende Kantenzug muss ebenfalls in der Triangulierung erhalten bleiben. Dadurch kann vermieden werden, dass bei einer späteren triangulierungs-basierten Interpolation über diese Linienobjekte hinweg interpoliert wird.

Die zusätzlichen Randbedingungen können durch eine Delaunay Triangulation einfach berücksichtigt werden. Bei der Erstellung der Delaunay Triangulierung wird lokale Optimalität überall angestrebt, solange die vorgegebenen Kanten dem nicht entgegen stehen. Dies wird dahingehend erreicht, dass die vorgegebenen Kanten immer als Randkanten der zu triangulierenden Vierecke verwendet werden.

In der nachfolgen Abbildung wird der Unterschied zwischen dem Ergebnis einer klassischen Delaunay Triangulierung und der bedingten Delaunay Triangulierung einer Punktmenge P mit einem nicht-konvexen Grenzpolygon dargestellt. Die rote Kante wurde durch die klassischen Delaunay Triangulierung und kreuzt die Grenzlinie des Grenzpolygons Ω. Anstelle der roten Kante berücksichtigt die grüne Kante das Grenzpolygon.

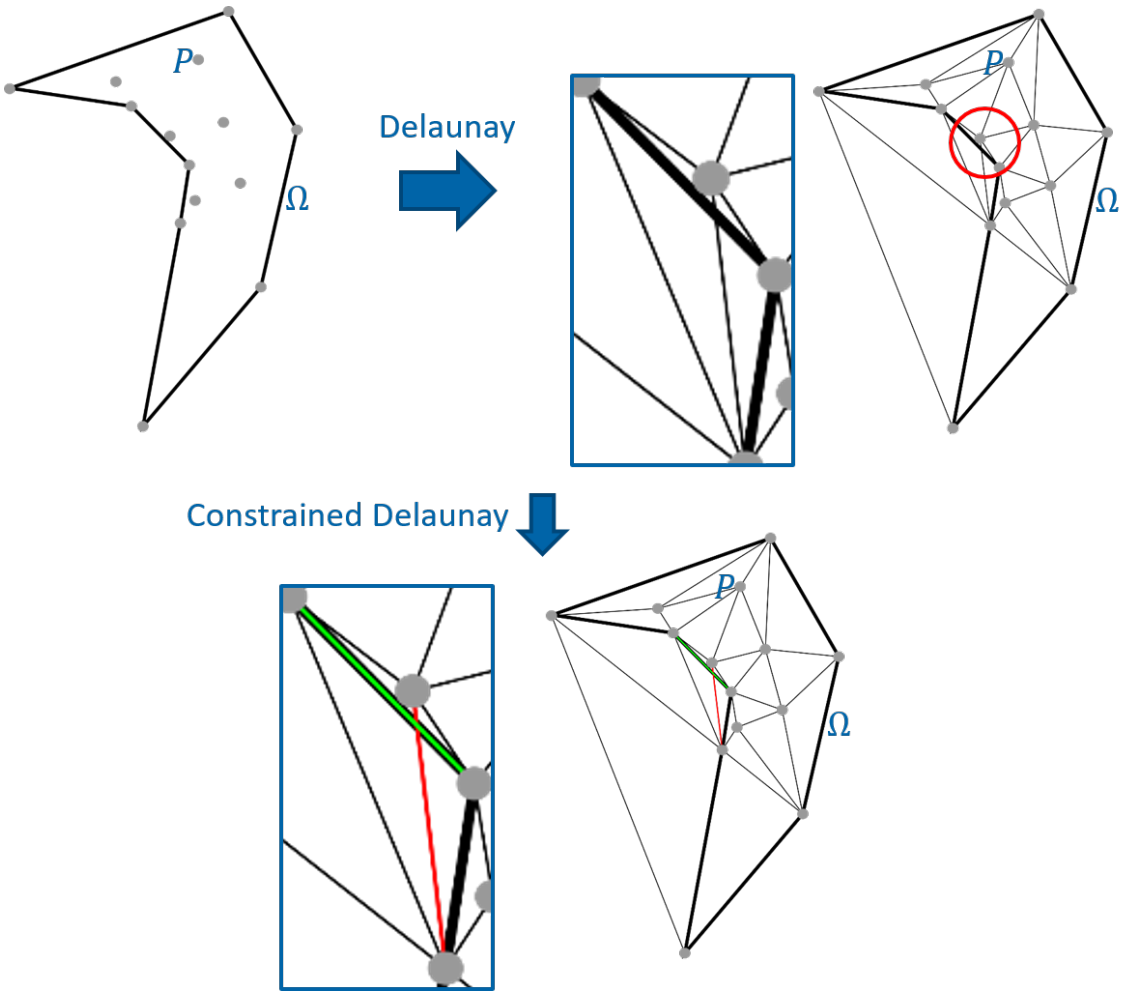


Abbildung 59: Durch eine bedingte Delaunay Triangulierung wird sichergestellt, dass alle Kanten des nicht-konvexen Grenzpolygons Ω der Punktmenge P in die Triangulierung mit einfließen

5.3 Vermaschung zwischen Linienobjeken

Ein häufiges Problem ist die Verbindung zweier benachbarten Linienobjekte durch Dreiecke. Wenn man zum Beispiel den möglichen Parameterwert zwischen zwei Isolinien (Linien gleichen Parameterwertes, auch Isohypsen genannt) vorhersagen möchte, ist es hilfreich, beide Linien mittels Dreiecken zu verknüpfen. Grundsätzlich gibt dafür viele verschiedene Ansätze. Die im Folgenden beschriebene Methode basiert auf dem in Meyers et al. 1992 beschriebenen Vorgehen (Meyers et al. 1992: Surfaces from Contours. ACM Transactions on Graphics, Vol. 11, No. 3, July 1992, Pages 228-258).

Seien C_i und C_j zwei Linienobjekte aus geordneten Sequenzen von aufeinanderfolgenden Punkten mit

$$C_i = \{p_1, p_2, \ldots, p_i, p_{i+1}, \ldots, p_{n_i}\}$$

und

$$C_j = \{q_1, q_2, \ldots, q_j, q_{j+1}, \ldots, q_{n_j}\}.$$

c_i ist ein lineares Segmente von C_i zwischen den Punkten p_i und p_{i+1}. Analoges gilt für ein Segment $c_j \in C_j$.

Eine solche Situation ist beispielhaft in der folgenden Abbildung dargestellt. Beiden Linien sollen durch Dreiecke verbunden werden. Die zugehörigen Punkte sind sortiert und durchnummeriert.

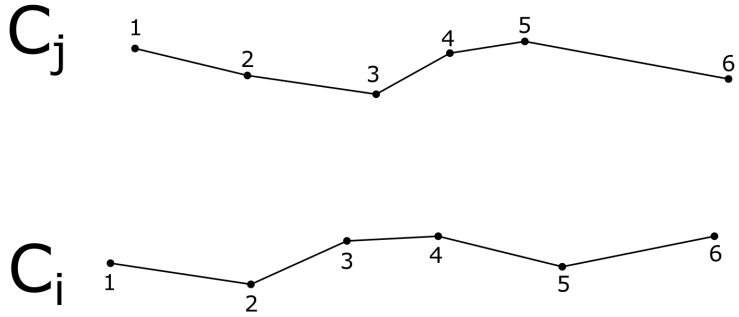


Abbildung 60: Beispielsituation mit zwei zu verbindenden Linienobjekten C_i und C_j

Zwei Linienobjekte lassen sich über eine so genannte **Parkettierung** (engl. *tiling*) über Dreiecke (engl. tiles) verknüpfen. Es handelt sich um einen speziellen Typ einer räumlichen (spatial) Triangulierung, die folgende Bedingungen erfüllen soll:

- Aufeinanderfolgende Punkte auf den Linienobjekten sollen direkt durch eine Dreieckskante verbunden werden. Dies bedeutet, dass alle Segmente c_i und c_j als Dreieckskanten in die Parkettierung mit einfließen.
- Es dürfen **keine ebenen** (*flat*) Dreiecke entstehen. Ebenen Dreiecke verwenden nur Punkte, welche alle zum gleichen Linienobjekt gehören. Die erstellten Dreieck müssen immer Punkte beider Linienobjekte umfassen.

Für die oben gezeigte Beispielsituation sind in der folgenden Abbildung mögliche ebene Dreiecke (rot) innerhalb der konvexen Hülle (rot, gestrichelt) dargestellt. Diese gilt es bei der Erstellung der Parkettierung zu vermeiden.

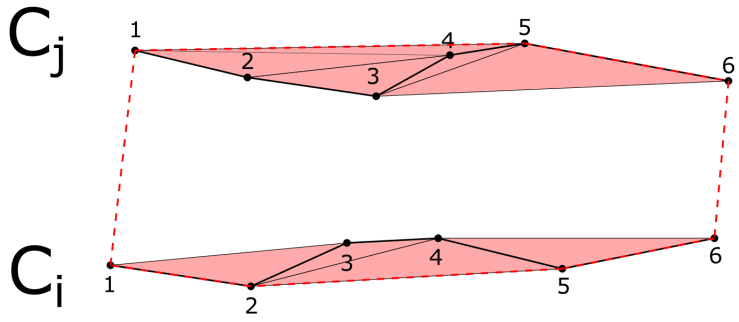


Abbildung 61: Ebene Dreieck (rot) innerhalb der konvexen Hülle (rot, gestrichelt) für die zu verbindenden Linienobjekte

Bei einer Parkettierung handelt es sich **nicht** um eine klassische ebene Triangulierung. Zum Einen überdeckt die Vereinigung aller erstellten Dreiecke meist nicht die gesamte konvexe Hülle der zu verknüpfenden Linienobjekte, zum Anderen kann nicht generell sichergestellt werden, dass sich die erzeugten Dreiecke nicht überlappen.

Eine Parkettierung kann über einen rechteckigen Suchgraph abgebildet werden. Die Zeilen entsprechen den Punkten $\{p_1, p_2, \ldots, p_i, p_{i+1}, \ldots, p_{n_i}\} \in C_i$ die Spalten den Punkten $\{q_1, q_2, \ldots, q_j, q_{j+1}, \ldots, q_{n_j}\} \in C_j$. Jeder Knoten im Graph entspricht einem Tupel (p_i, q_j) und repräsentiert eine mögliche Kante zwischen den beiden Linienobjekten. In der nachfolgenden Abbildung ist der Suchgraph für die Beispielsituation skizziert. Die möglichen Knoten sind als graue Punkte dargestellt. Der rote/orange Knoten entspricht der roten/orangen Kante in der Parkettierung. Die Verbindung $(p_i, q_j) \rightarrow (p_k, q_l)$ zweier Knoten (p_i, q_j) und (p_k, q_l) entspricht dem Dreieck, welches die beiden verbundenen Knoten/Kanten (p_i, q_j) und (p_k, q_l) verwendet. Die dritte Kante ergibt sich implizit. Ein Beispiel hierfür ist das gelbe Dreieck / die gelbe Knotenverbindung $(p_1, q_1) \rightarrow (p_1, q_2)$ in der nachfolgenden Abbildung.

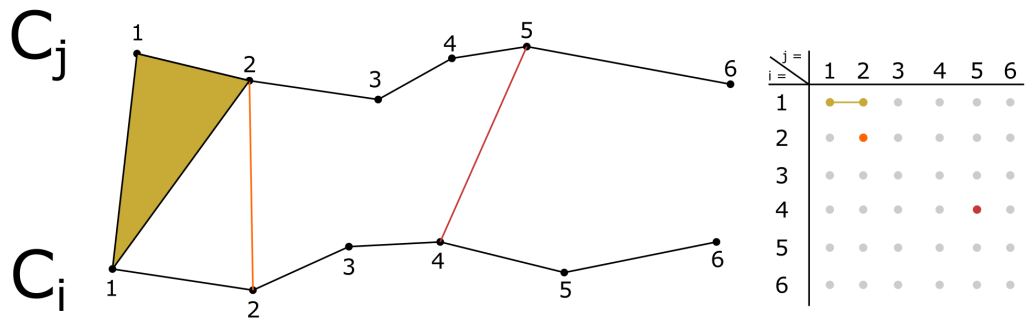


Abbildung 62: Suchgraph (rechts) zur Erstellung der Parkettierung des Beispiels (links)

Um die Parkettierung zu erhalten, muss im Suchgraph ein Pfad (aufeinanderfolgende Sequenz von Knotenverbindungen) ausgehend vom Knoten (p_1, q_1) bis zum Knoten (p_{n_i}, q_{n_j}) aufgespannt werden. Dafür sind nur Verbindungen zwischen direkt benachbarten Knoten erlaubt. Ein Knoten (p_i, q_j) darf nur entweder mit dem Knoten (p_{i+1}, q_j) oder dem Knoten (p_i, q_{j+1}) verbunden werden. In den folgenden beiden Abbildungen sind mögliche Pfade und die resultierenden Parkettierungen für die Beispielsituation dargestellt. Beide Parkettierungen (gelb und rot) weisen die gleiche Anzahl von Knotenverbindungen und damit Dreiecken auf. Die Form der Dreiecke unterscheidet sich allerdings erheblich.

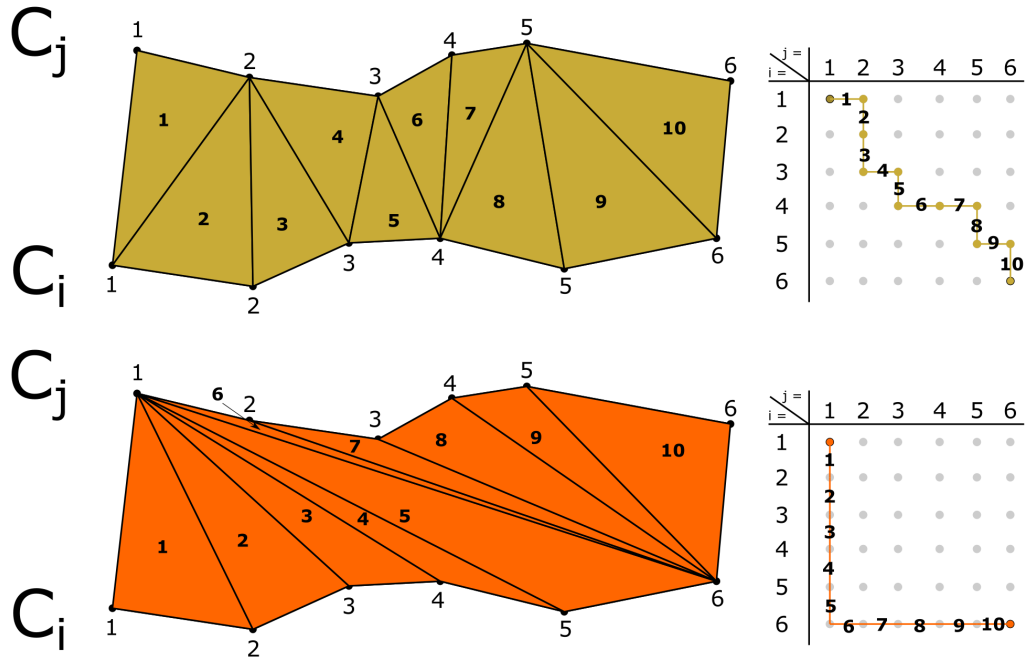


Abbildung 63: Mögliche Parkettierungen zur Beispielsituation und zugehörige Pfade im Suchgraph

Eine eindeutige Parkettierung kann konstruiert werden, in dem der "beste / optimalePfad in diesem Suchgraph gefunden wird. Der optimale Pfad hängt immer von einem gewählten Kriterium ab, welches die Güte der Parkettierung beschreibt. Es ist grundsätzlich möglich, dass mehrere optimale Pfade im Suchgraph existieren, dann lässt sich zwar keine eindeutige Parkettierung mehr aufbauen, die möglichen Parkettierungen können als als äquivalent betrachtet werden.

Für die oben gezeigt Beispielsituation wird die Gesamtfläche aller Dreiecke als Optimierungskriterium angenommen, welches es zu minimieren gilt. Dafür wird jede Knotenverbindung im Suchgraph gemäß der Fläche des repräsentierten Dreiecks gewichtet. Diese Gewichte sind in der nachfolgenden Abbildung (links) dargestellt. Diese Gewichte lassen sich auch als Länge"der Verbindung zwischen zwei Knoten im Suchgraph interpretieren. Das kumulative Gewicht einer Verbindung ausgehend von einem Knoten (p_i, q_j) ist die Summe des Gewichtes der Verbindung mit dem kumulativen Gewicht einer der beiden Verbindungen, welche zum Ausgangsknoten der Verbindung führen, $(p_{i-1}, q_j) \rightarrow (p_i, q_j)$ oder $(p_i, q_{j-1}) \rightarrow (p_i, q_j)$. Es wird dabei immer das **kleinere kumulative Gewicht** der vorhergehenden Verbindungen genutzt. Das kumulative Gewicht einer Verbindung $(p_i, q_{j-1}) \rightarrow (p_i, q_j)$ lässt sich als die Gesamtlänge des Pfades bis zum Erreichen des Knoten (p_i, q_j) interpretieren. Die kumulativen Gewichte für den Beispielsachverhalt sind in der nachfolgenden Abbildung (rechts) dargestellt.

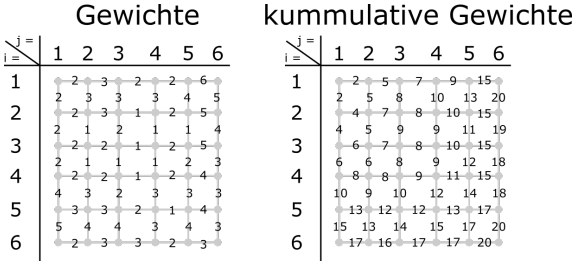


Abbildung 64: Gewichte und kumulative Gewichte für alle möglichen Knotenverbindungen

Der optimale Pfad ausgehend vom Knoten (p_1, q_1) bis zum Knoten (p_{n_i}, q_{n_j}) ergibt sich als der Pfad mit der "kürzesten"kumulativen Länge. Er kann aufgebaut werden, in dem der Suchgraph ausgehend von Zielknoten (p_{n_i}, q_{n_j}) rückwärts durchlaufen wird. Ausgehend von einem Knoten wird dabei die Verbindung zu einem vorherigen Knoten gewählt, welche das kleinere kumulative Gewicht aufweist. Dies ist für den Beispielsachverhalt in der folgenden Animation dargestellt. Hier treten manchmal Fälle auf, in denen beide zur Wahl stehende Verbindungen das gleiche kumulative Gewicht aufweisen. In diesem Fällen wird die vertikale Verbindung $(p_i, q_j) \rightarrow (p_{i-1}, q_j)$ bevorzugt.

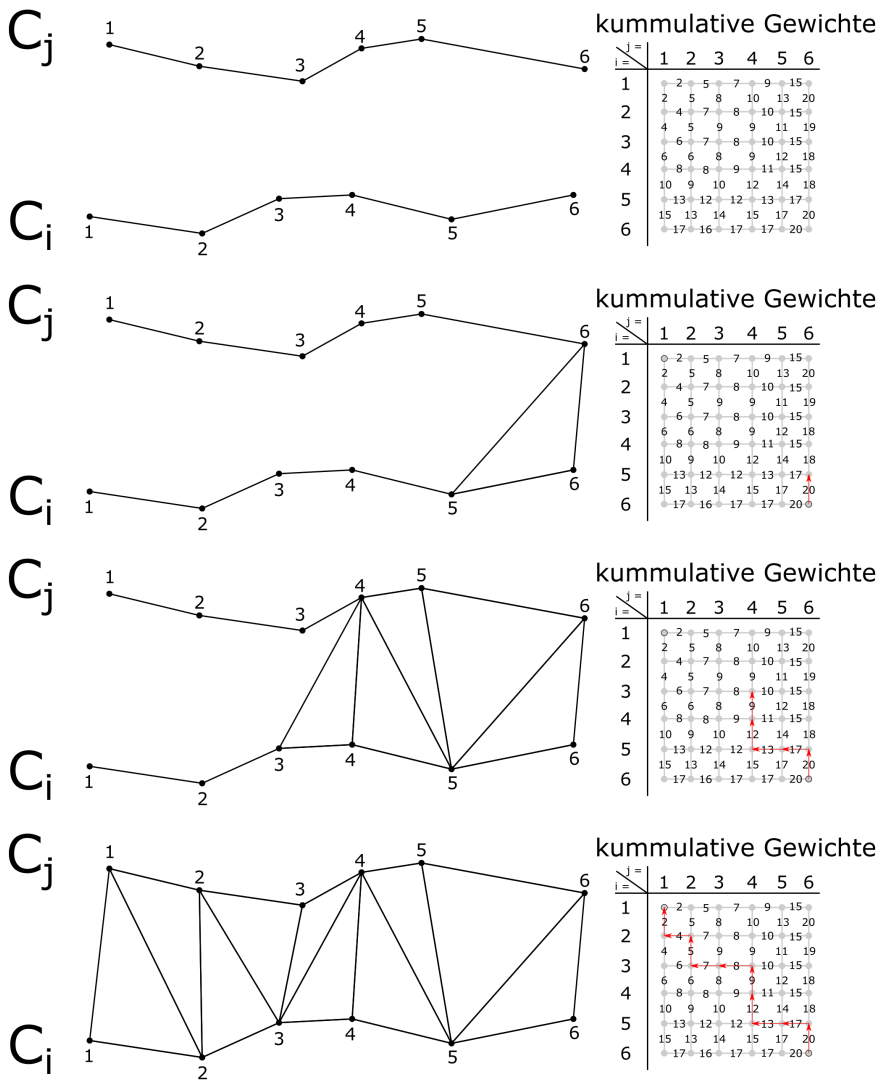


Abbildung 65: Aufbau der Parkettierung basierend auf dem optimalen Pfad im Suchgraph

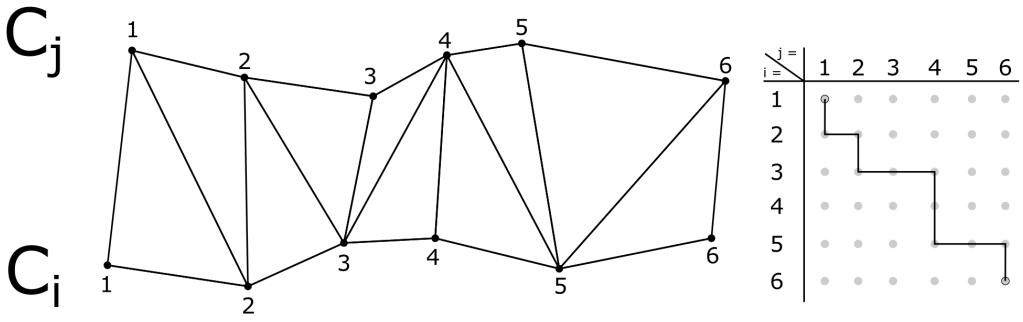


Abbildung 66: Optimale Parkettierung der Beispielsituation

Im Gegensatz zur (bedingten) Delaunay Triangulation werden durch die Parkettierung nicht notwendiger Weise optimale Dreiecke gemäß des Max-Min-Kriteriums erzeugt, dafür weist das oben gezeigte Ergebnis die minimale Gesamtfläche auf. Es beinhaltet zudem keine ebenen Dreiecke, welche bei einer Delaunay Triangulierung durchaus auftreten können. Bei der Parkettierung ist es des Weiteren nicht notwendig, nachträglich Dreiecke zu entfernen, welche zwar innerhalb der

konvexen Hülle der Linienobjekte, aber nicht zwischen beiden Linienobjekten liegen.

Durch eine solche Parkettierung ist eine Interpolation (siehe Abschnitt zur *Linearen Interpolation auf Dreiecken*) zwischen Isolinien sehr leicht möglich. Zudem lassen sich die entstehenden Dreieckstreifen (*triangle strips*) in vielen Visualisierungssystemen effizienter darstellen als zum Beispiel eine Delaunay Triangulierung.

6 Räumliche Vorhersage / Interpolation

Eine der wichtigsten Anwendungen eines GIS ist die Vorhersage von Sachverhalten basierend auf den bekannten Daten in der GIS-Datenbank. Diese Vorhersage basiert unter anderem auf der Datenanalyse und dient ebenfalls zur Entscheidungsfindung.

Bei der räumlichen Vorhersage ist das Ziel, ein an wenigen Lokationen bekanntes Attribut an Positionen vorherzusagen, an denen es initial unbekannt ist. Diese räumliche Vorhersage wird allgemein auch als **Interpolation** bezeichnet. Eine sehr häufige Anwendung für Interpolation ist die Übertragung von Werten für ein Attribut, das nur an Einzelpunkten vorliegen, auf die Fläche (Raster, Polygone), z.B. zur Erzeugung von Kartendarstellungen für dieses Attribut.

Ein praktisches Beispiel ist hier die Erstellung eines digitalen Geländemodells (DGM). Dieses soll die Höheninformationen möglichst als regelmäßiges Raster bereitstellen. Zumeist kann die "wahre"Höheninformation allerdings weder so regelmäßig, noch in der notwendigen Auflösung tatsächlich gemessen werden. Die "wahren" Höhenpunkte liegen als zumeist unregelmäßig und mit zum Teil hohen Abständen vor. Über Interpolation kann die bekannte Höheninformation nun auf die regelmäßig verteilten Punkte des DGM übertragen werden. Ein Beispiel dafür ist in der folgenden Abbildung dargestellt. Je nach verwendetem Interpolationsansatz unterscheiden sich die Ergebnisse.

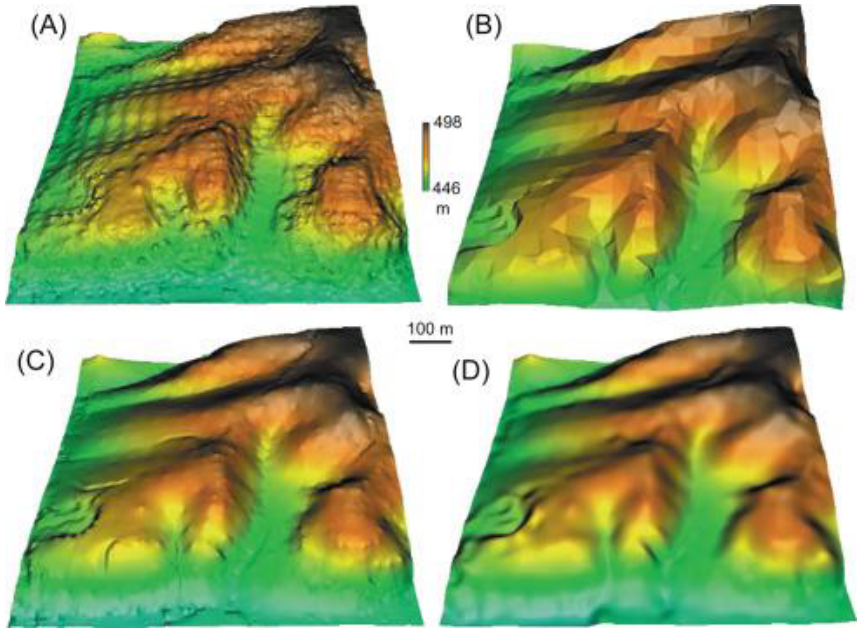


Abbildung 67: Beispiele für ein über Interpolation erzeugtes digitales Geländemodell. Es wurden verschiedene Interpolationsansätze verwendet: IDW (A), Lineare Interpolation auf Dreiecken (B), Splineinterpolation (C), Splineapproximation (D)

6.1 Grundlagen

In der numerischen Mathematik bezeichnet der Begriff *Interpolation* eine Klasse von mathematischen Verfahren und Problemen. Diese befassen sich immer damit, zu einer gegebenen Menge an Daten eine stetige Funktion zu finden, welche diese Daten abbildet. Die Daten werden dann über dieser Funktion *interpoliert*. Zudem lassen sich mit dieser Funktion unbekannte Werte für neue Positionen vorhersagen. Im Kontext von GIS soll also ausgehend von bekannten Datenwerten an bekannten Positionen eine mathematische Vorschrift gefunden werden, welche eine geowissenschaftlich plausible Vorhersage von unbekannten Werten an Positionen ohne gegebene Werte erlaubt. Diese Funktionen oder Vorschriften werden auch als **Modelle** für die Interpolation bezeichnet. Sie formalisieren die Modellvorstellungen, wie sich das zu interpolierende Attribut ausgehend von den bekannten Werten / Positionen im Raum fortsetzen könnte. Die so vorhergesagten Werte basieren auf dieser Modellvorstellung. Sie können die realen, aber unbekannten Werte gegebenen Falls

plausibel repräsentieren, müssen aber immer als Schätzung ode Näherung an diese unbekannten Werte betrachtet werden.

Grundsätzlich lassen sich nur Attribute für kontinuierliche Variablen über Interpolation vorhersagen, da für die Interpolation stetige Funktionen verwendet werden, d.h. es können im Allgemeinen unendlich viele verschiedene Werte für ein Attribut vorhergesagt werden. Das Ergebnis einer Interpolation ist klassischer Weise eine kontinuierliche Parameterverteilung. Dies ist nicht sinnvoll für diskrete Attribute, da diese auf eine begrenzte Menge von Attributwerten beschränkt sind. Diskrete Attributwerte, wie zum Beispiel eine Lithologie, lassen sich nicht sinnvoll interpolieren, da sich zwischen einer Klasse A und einer Klasse B keine kontinuierlichen Zwischenklassen bestimmen lassen. Eine Ausnahme bildet die *Nearest-Neighbor-* oder auch *stückweise-konstante*-Interpolation. Diese erlaubt es, dass die interpolierten Werte nur die Werte annehmen, welche durch die gegebenen Datenwerte bereits bekannt sind. So können auch diskrete Parameterverteilungen vorhergesagt werden. Die Vorhersage von diskreten Parameterwerten erfolgt klassischer Weise über Verfahren der Klassifikation, welche nicht Teil dieser Veranstaltung sind.

In der nachfolgenden Animation sind Beispiele für die Vorhersage / Interpolation zweier flächenhafter Schichten im Untergrund entlang eines vertikalen Profils dargestellt. Die Geometrie der Schichten ist jeweils nur an drei Bohrmarkern bekannt (vertikale Linien). Einfache mathematische Modelle führen zu sehr einfachen und vermutlich unrealistischen Schichtverläufen. Realistischere Geometrien sind nur durch sehr komplizierte Modelle und eventuelle Zusatzinformationen (z.B. Störung; grau, gestrichelt) zu erreichen.

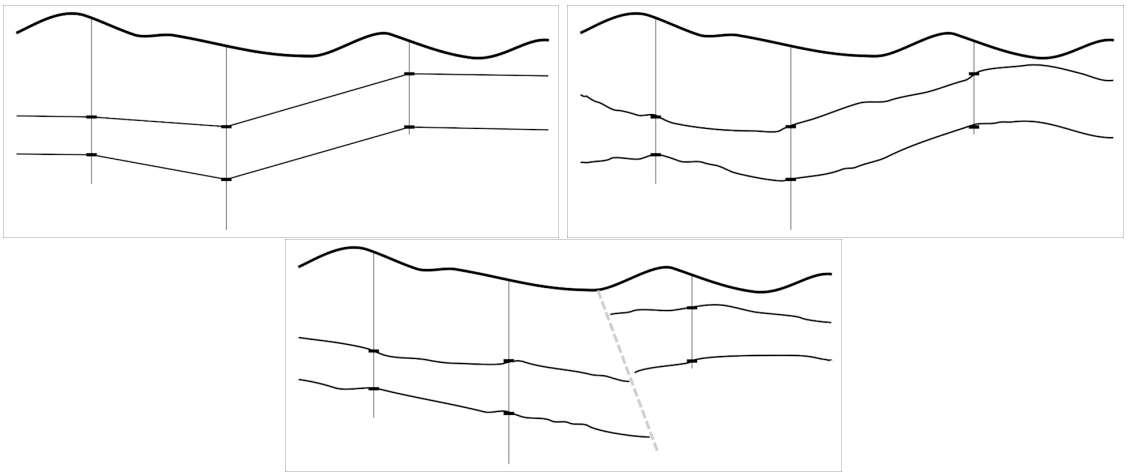


Abbildung 68: Beispiele für die Interpolation der Geometrie zweier flächenhaften Schichten im Untergrund (dicke schwarze Linie: Topografie). Es ist hier jeweils der Verlauf der Schichten auf einem vertikalen 2D Profil dargestellt

Mathematische Grundlagen

Gegeben sei eine Punktmenge P, für jeden Punkt $p_i \in P$ mit $i = 1, \ldots, m$ liegt neben seiner Position $\vec{x_i} \in D \subset R^n$ auch ein bekannter Wert $f_i \in R$ für einen Parameter / ein Attribut vor mit $p_i = (\vec{x_i}, f_i)$. Der Unterraum D des Raums R^n definiert das Gebiet, indem die Interpolation gültig ist bzw. statt finden soll und könnte der R^n selbst oder zum Beispiel die konvexe Hülle von P sein mit $D = [P]$. f_i kann als "Wert an der Position $\vec{x_i}$" $f(\vec{x_i})$ oder äquivalent als "Wert am Punkt p_i" $f(p_i)$) aufgefasst werden.

Gesucht wird eine Rechenvorschrift $\hat{f}$, welche jeder Position $\vec{x} \in D$ einen eindeutigen Wert $\hat{f}(\vec{x})$ zuordnet. $\hat{f}$ ist das **Modell** der Interpolation und bildet die Modellvorstellung zur wahren, aber unbekannten Natur $\bar{f}$ des zu interpolierenden Attributs ab.

Dies wird in der nachfolgenden Abbildung schematisch für den oben animierten Sachverhalt gezeigt. Die Ausgangslage ist links dargestellt. Für zwei grundsätzlich unbekannte Sachverhalte $\bar{f_1}$ und $\bar{f_2}$ sind an drei Positionen die Werte $f_1(\vec{x_1})$ bis $f_2(\vec{x_3})$ bekannt. $\hat{f_1}$ und $\hat{f_2}$ sind mögliche Modelle für $\bar{f_1}$ und $\bar{f_2}$.

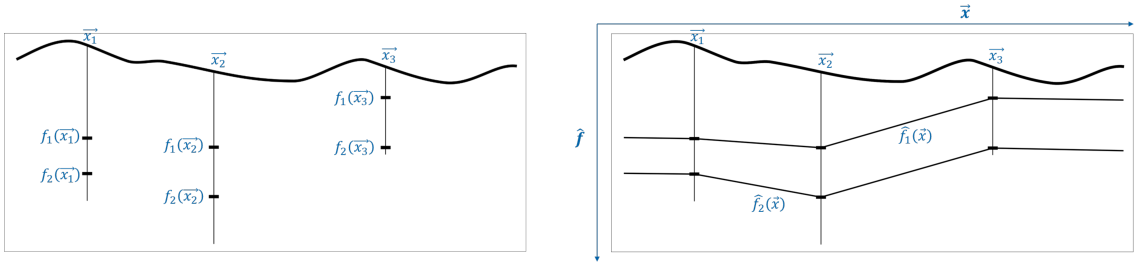


Abbildung 69: Schematische Darstellung für ein Interpolationsproblem

Im Allgemeinen lässt sich jede klassische Interpolation als gewichtetes Mittel

$$\hat{f}(\vec{x}) = \sum_{i}^{m} w_i(\vec{x}) f(\vec{x_i})$$

an der Position $\vec{x}$ über die bekannten Werte an allen Datenpositionen formulieren. Die Gewichte $w_i(\vec{x})$ werden gemäß des gewählten Modells bestimmt.

Exakte und nicht-exakte Interpolation (Approximation)

Für eine **exakte Interpolation** (manchmal auch als Interpolation im engeren Sinne bezeichnet) gilt in jedem Fall

$$\hat{f}(\vec{x_i}) = f(\vec{x_i}),$$

das heißt, die interpolierende Funktion $\hat{f}$ rekonstruiert die bekannten Werte an den Positionen $\vec{x_i}$ immer **exakt**.

Eine **nicht-exakte Interpolation** rekonstruiert die bekannten Werte an den Positionen $\vec{x_i}$ nur näherungsweise mit

$$\hat{f}(\vec{x_i}) \approx f(\vec{x_i}).$$

Die gegebenen Werte werden nur approximiert, also angenähert. Verfahren dieser Art werden deshalb auch als **Approximation** bezeichnet. Bei einer Approximation wird davon ausgegangen, dass auch die bekannte Werte $f(x_i)$ nicht exakt den wahren Werten $\bar{f}(\vec{x_i})$ entsprechen. Dies kann unter Berücksichtigung eines unbekannten Fehlers $\epsilon \neq 0$ wie folgt formuliert werden:

$$f(\vec{x_i}) = \bar{f}(\vec{x_i}) + \epsilon$$

Da der Fehler unbekannt ist, wird hier davon ausgegangen, dass eine exakte Rekonstruktion der bekannten Messwerte nicht notwendig ist.

Dies ist schematisch für den in der obigen Animation bereits gezeigten Sachverhalt in der nachfolgenden Abbildung dargestellt. Links werden die bekannten Werte exakt interpoliert, rechts nur approximiert. Die roten Marker stellen die approximierten Werte $\hat{f}(\vec{x_i})$ der bekannten Werte $f(x_i)$ (schwarz) dar.

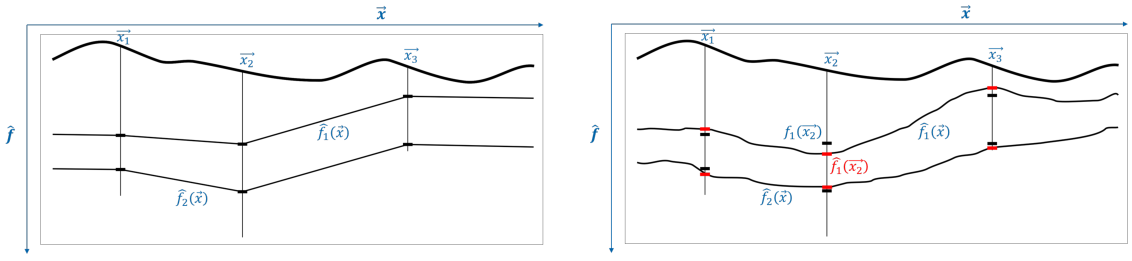


Abbildung 70: Schematische Darstellung des Unterschieds zwischen exakter Interpolation (links) und Approximation (rechts)

Globale und lokale Interpolation

Bei einer **globalen** Interpolation werden immer alle bekannten Werte $f(\vec{x_i})$ für die Interpolation an einer Position $\vec{x}$ berücksichtigt. Dies bedeutet, dass für alle Gewichte grundsätzlich gilt:

$$w_i(\vec{x}) \neq 0, \forall \vec{x_i}.$$

Im Gegensatz dazu werden bei einer **lokalen Interpolation** nur die Werte $f(\vec{x_i})$, welche sich innerhalb einer Nachbarschaft $N(\vec{x}$ um den zu interpolierenden Punkt $\vec{x}$ befinden, für die Interpolation verwendet. Für die Interpolationsgewichte gilt:

$$w_i(\vec{x}) \begin{cases} \neq 0 & \forall \vec{x_i} \in N(\vec{x}) \\ = 0 & \forall \vec{x_i} \notin N(\vec{x}) \end{cases}.$$

Die Bestimmung der Nachbarschaften $N(\vec{x}$ hängt dabei vom verwendeten Verfahren ab.

Dies ist schematisch in der nachfolgenden Abbildung dargestellt. Links stellt eine globale Interpolation dar, links eine lokale Interpolation. Nur die grauen, schwarz umrandeten Datenpunkte werden für die Interpolation berücksichtigt.

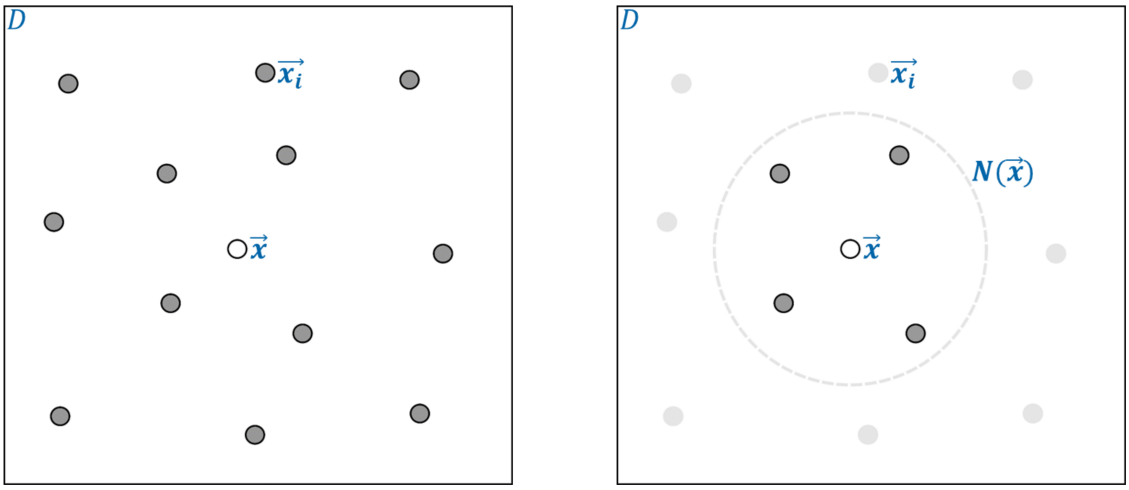


Abbildung 71: Unterschied zwischen globaler (links) und lokaler (rechts) Interpolation

Deterministische und stochastische Interpolation

Grundsätzlich kann für jede Interpolation angenommen werden, dass sich die Gewichte $w_i(\vec{x})$ für die Datenwerte mit zunehmenden Abstand zum Interpolationspunkt verringern. Weit entfernte Daten sollten einen geringeren Einfluss auf eine zu interpolierende Position haben als nahe Datenpunkte. Die Gewichte hängen häufig in irgendeiner Form vom Abstand des Interpolationspunktes zum i-ten Datenpunkt ab mit $w_i(\vec{x}) = w_i(d(\vec{x} - \vec{x_i}))$ und werden meist mit steigendem Abstand immer kleiner. Die Interpolationsgewichte lassen sich sowohl über **deterministische** als auch über **stochastische** Ansätze bestimmen.

Bei einer **deterministischen** Interpolation werden die Gewichte rein basierend auf einer durch das mathematische Modell bestimmten Gewichtsfunktion ermittelt.

Bei einer **stochastischen** oder auch **geostatistischen** Interpolation basiert die Gewichtsbestimmung zusätzlich auf einer statistischen Auswertung der bekannten Datenwerte. Solche statistischen Modelle berücksichtigen zum Beispiel die räumliche Korrelation zwischen den bekannten Datenwerten. Dadurch lassen sich die interpolierten Werte nicht nur zuverlässiger vorhersagen als mit deterministischen Verfahren, sondern häufig kann zudem ein Maß für die Unsicherheit der Vorhersage angegeben werden.

Extrapolation

Eine zuverlässige Interpolation kann nur **innerhalb** der bekannten Daten durchgeführt werden. Hier können die Vorhersagen des Modells als plausibel angenommen werden.

Auch für Zielpunkte **außerhalb** der Daten, die zum Teil weit von den gegebenen Datenlokationen entfernt sind, lässt sich häufig über ein Interpolationsmodell ein Wert vorhersagen. Dies wird als **Extrapolation** bezeichnet. **Extrapolierte Wert müssen immer kritisch gesehen werden**, da die Vorhersage nur aufgrund von ungenügenden Daten erfolgt. Abhängig vom verwendeten Modell sind die extrapolierten Werte häufig nicht mehr plausibel für die Realität. Manche Interpolationsmodelle erlauben zudem rein mathematische keine Extrapolation.

In der nachfolgenden Abbildung ist dies schematisch für eine 1D Situation nochmals verdeutlicht. Nur im Bereich zwischen den Datenpunkten ist eine zuverlässige Interpolation möglich. Außerhalb der Daten findet eine Extrapolation statt.

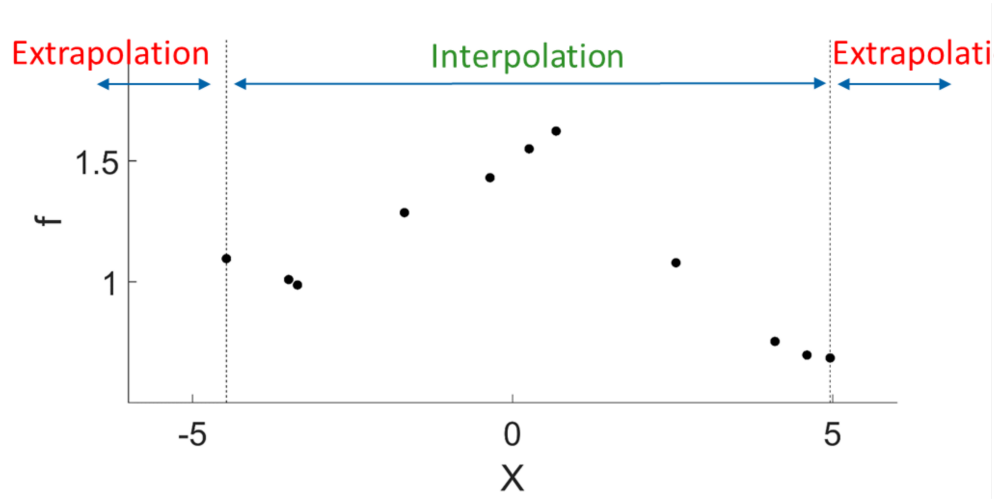


Abbildung 72: Regionen, in denen für gegebene 1D Datenpunkte Interpolation und Extrapolation stattfindet

6.2 Deterministische Interpolation von verteilten Punktdaten

Im Folgenden werden verschiedene, häufig verwendete deterministische Interpolationsverfahren vorgestellt, welche keine wie auch immer geartete Diskretisierung voraussetzen. Diese Verfahren arbeiten auf einer Menge P von unterschiedlichen Punkten $p_i = (\vec{x_i}, f_i) \in P$ mit $i = 1, \ldots, m$ und benötigen **keine Vermaschung** dieser Punkte.

Jedes dieser Verfahren beruht auf einer anderen Modellannahme. Sie lassen sich also für unterschiedlich Sachverhalte unterschiedlich gut verwenden. Die Natur des Sachverhalts sollte immer durch die Modellannahme berücksichtigt werden, um eine plausible Interpolation durchführen zu können.

Die Verfahren werden an dem unten gezeigten 1D Beispiel mit 11 Punkten $p_i = (x_i, f_i)$ mit $i = 1, \ldots, 11$ demonstriert, können im Allgemeinen aber auf höhere Dimensionen erweitert werden. Vor allem in den Bereichen, wo Extrapolation stattfindet, unterscheiden sich die Vorhersageergebnisse erheblich.

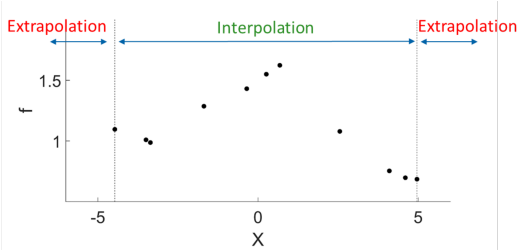


Abbildung 73: Beispieldaten in 1D zur Demonstration der verschiedenen Interpolationsverfahren. Die Regionen, in denen Interpolation oder Extrapolation stattfindet, sind gesondert gekennzeichnet

Inverse Distanzwichtung (*inverse distance weighting* - IDW)

Bei der **IDW-Interpolation** wird die Annahme, dass nahe Datenpunkte einen höheren Einfluss auf einen Interpolationspunkte haben als weiter entfernte Punkte, direkt zur Berechnung der interpolierten Werte an den Interpolationspunkten verwendet. Es handelt sich um ein sehr einfaches Verfahren, dessen Modellannahme in vielen Fällen grundsätzlich plausibel ist, aber häufig komplexere Strukturen nicht adequat abbildet.

Modellannahme: Ein vorhergesagter Wert für eine Position $\vec{x}$ wird von allen bekannten Werten $f_i = f_i(\vec{x_i})$ beeinflusst. Der Einfluss nimmt proportional zum Abstand $d(\vec{x}, \vec{x_i})$ ab.

Für eine Position $\vec{x}$ wird der Wert $\hat{f}(\vec{x})$ über die folgende Gleichung berechnet:

$$\hat{f}(\vec{x}) = \frac{\sum_{i=1}^{m} w_i(\vec{x}) f_i}{\sum_{i=1}^{m} w_i}$$

Jedes Gewicht $w_i(\vec{x})$ entspricht der inversen Distanz zwischen den Positionen $\vec{x_i}$ und $\vec{x}$ mit

$$w_i(\vec{x}) = \frac{1}{d(\vec{x}, \vec{x_i})^k}.$$

k ist ein so genannter Lokalisierung-Parameter, der beschreibt, wie stark das Gewicht mit zunehmendem Abstand geringer wird. In den meisten Fällen wird $k = 2$ gewählt. Bei IDW handelt es sich immer um einen **exakten** Interpolator.

Die folgenden Abbildungen zeigen die IDW-Interpolationsergebnisse für das 1D Beispiel für verschiedene Lokalisierungs-Parameter. Die Ergebnisse unterscheiden sich erheblich. Für das Beispiel wäre die Wahl von $k = 2$ angemessen.

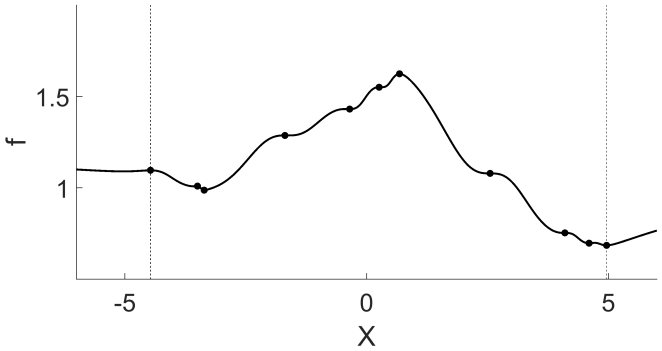


Abbildung 74: **IDW-Interpolation mit** $k = 2$

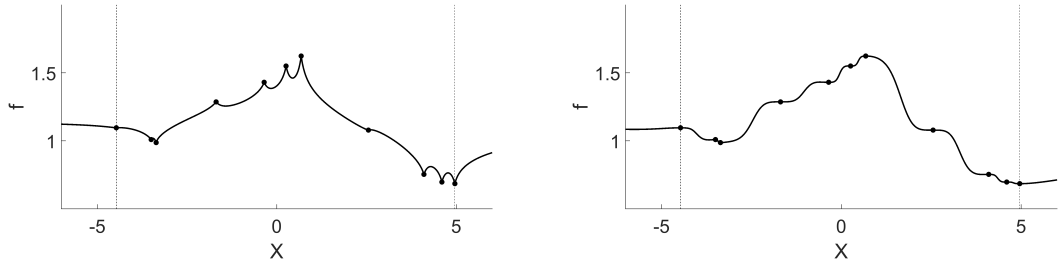


Abbildung 75: IDW-Interpolation mit $k = 1$ (links) und mit $k = 3$ (rechts)

Polynomapproximation

Bei der **Polynomapproximation** wird versucht, einen Sachverhalt $\hat{f}$ über eine polynomiale Funktion abzubilden. Für m bekannte Punkte $p_i = (x_i, f_i)$ existiert genau ein Polynom vom Grad $k = m - 1$, welches diese Punkte exakt abbildet.

Im 1D lässt sich ein allgemeines Polynom eines beliebigen Grades k wie folgt darstellen:

$$\mathcal{P}(x, k) = \sum_{j=0}^{k} a_j x^j.$$

Ein Polynom 3. Grades im 1D hätte also die Form

$$\mathcal{P}(x, 3) = a_3 x^3 + a_2 x^2 + a_1 x + a_0.$$

Im 2D sind die Polynome komplexer, weil zur x-Koordinate noch die y-Koordinate hinzugenommen werden muss. Ein Polynom 2. Grades im 2D hätte die Form

$$\mathcal{P}(x, y, 2) = a_5 x^2 + a_4 y^2 + a_3 xy + a_2 x + a_1 y + a_0.$$

Polynome haben den Vorteil, dass sie sich analytisch abbilden lassen und an jeder Position $\vec{x}$ beliebig oft stetig differenzierbar sind. Sie neigen allerdings bei hohen Graden k zu sehr starken Oszillationen. Eine exakte Abbildung für eine größere Menge an Punkten ist daher häufig unsinnig. Aus diesem Grund wird im Allgemeinen nicht erwartet, dass ein Polynom alle Datenwerte exakt repräsentiert. Es soll hingegen das Polynom für einen möglichst geringen Grad $k \ll m$ gefunden werden, bei dem die Abweichungen zu den bekannten Daten minimal sind. Es handelt sich also zumeist um eine **Approximation**.

Modellannahme: Ein vorhergesagter Wert für eine Position $\vec{x}$ wird lässt sich über ein Polynom $\mathcal{P}$ vom Grad $k \ll m$ darstellen. Dieses Polynom soll den Ausdruck

$$\sum_{i=1}^{m} (f_i(\vec{x_i}) - \mathcal{P}(\vec{x_i}, k))^2$$

minimieren.

Die notwendigen Koeffizienten $a_0, a_1, \dots a_l$ für das minimierende Polynom lassen sich über die Lösung eines Gleichungssystems $A\mathbf{x} = \mathbf{b}$ mit

$$A = \begin{bmatrix} x_1^l & x_1^{l-1} & \cdots & x_1^0 \\ x_2^l & x_2^{l-1} & \cdots & x_2^0 \\ \vdots & \vdots & \ddots & \vdots \\ x_m^l & x_m^{l-1} & \cdots & x_m^0 \end{bmatrix} \text{ (für 1D), } \mathbf{x} = \begin{bmatrix} a_l \\ a_{l-1} \\ \vdots \\ a_0 \end{bmatrix} \text{ und } \mathbf{b} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}$$

bestimmen. Dies ist eindeutig möglich (über die Methode der kleinsten Quadrate"), wenn gilt: $l+1 < m$; das heißt, wenn mehr bekannte Werte vorhanden sind, als Polynomkoeffizienten bestimmt werden müssen.

Die folgenden Abbildungen zeigen die Approximationsergebnisse für das 1D Beispiel für die Grade $k = 3$ und $k = 5$. Zusätzlich ist das Ergebnis für den Grad $k = 10$ abgebildet, welches die 11 gegebenen Daten exakt interpoliert, allerdings im extrapolierten Bereich sehr schnell extreme und damit unplausible Werte annimmt.

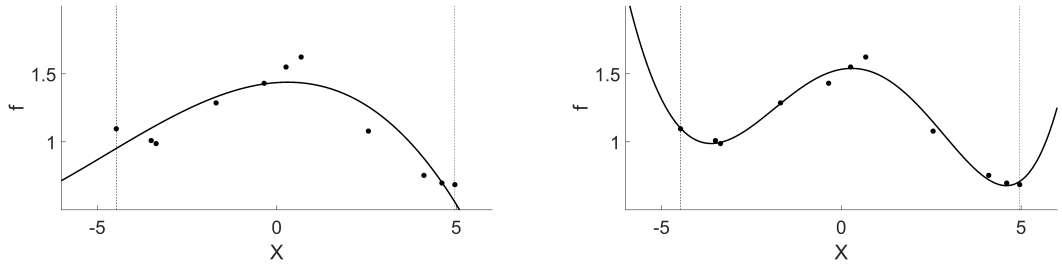


Abbildung 76: Polynomapproximation mit Grad $k = 3$ (links) und mit $k = 5$ (rechts)

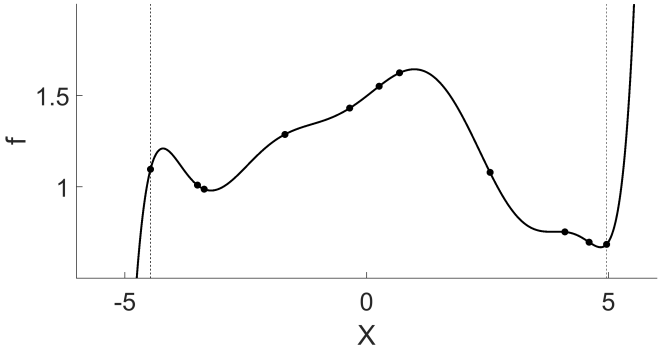


Abbildung 77: Polynominterpolation mit Grad $k = 10$

Interpolation über polynomiale Splines

m Datenwerte lassen sich bekanntlich exakt über ein Polynom vom Grad $m - 1$ abbilden. Dies ist allerdings nur für Punktmengen mit sehr wenigen Punkten (z.B. $m \leq 5$) sinnvoll durchführbar. Im Fall von sehr vielen Punkten ist es jedoch zumindest möglich, eine Funktion zu konstruieren, welche lokal, also zwischen den Datenpunkten in einem sehr begrenzten Gebiet, mit einem Polynom niedrigen Grades übereinstimmt. Dieses Polynom interpoliert diese (wenigen) Datenpunkte in diesem Gebiet exakt. Global lässt sich diese Funktion aber nicht mehr als einfaches Polynom beschreiben. Eine solche Funktion wird als **polynomialer Spline** bezeichnet. Ein polynomialer Spline vom Grad k ist im Gegensatz zu einem Polynom vom Grad k nicht mehr überall beliebig oft, sondern nur noch $(k-1)$-mal stetig differenzierbar. Das bedeutet, es lassen sich für diese Funktion global nur noch stetige Ableitungen bis zur Ordnung $(k-1)$ bestimmen. Da jedoch alle bekannten Daten exakt interpoliert werden, werden diese Splines in der Praxis der Polynomapproximation vorgezogen. Allerdings ist die Berechnung solcher Splines auch aufwändiger als die Berechnung von Approximationspolynomen niedrigen Grades.

Als Splines werden im allgemeinen Klassen von Funktionen bezeichnet, welche die Rauheit" bezüglich eines Sachverhaltes minimieren. Das Ziel ist es immer eine stückweise stetige Funktion zu erhalten, die möglichst glatt ist.

Ein polynomialer Spline vom Grad k zu einer Punktmenge P ist eine Funktion,

- die zwischen den Datenpunkten mit einem Polynom vom Grad k übereinstimmt und
- deren Ableitung bis einschließlich Ordnung $(k-1)$ stetig sind. Ableitungen höherer Ordnung sind **in den Datenpunkte**n unstetig.

Modellannahme: Ein vorhergesagter Wert $\hat{f}\vec{x}$ für eine Position $\vec{x}$ lässt sich lokal über ein Polynom niedrigen Grades bestimmen, jeder Datenpunkt muss dabei aber exakt reproduziert werden. $\hat{f}$ muss jedoch nicht überall beliebig oft stetig differenzierbar sein.

Die folgenden Abbildungen zeigen die Interpolationsergebnisse für das 1D Beispiel für verschiedene kubische Splines. kubische Splines entsprechen zwischen den Datenpunkten einem Polynom vom Grad 3.

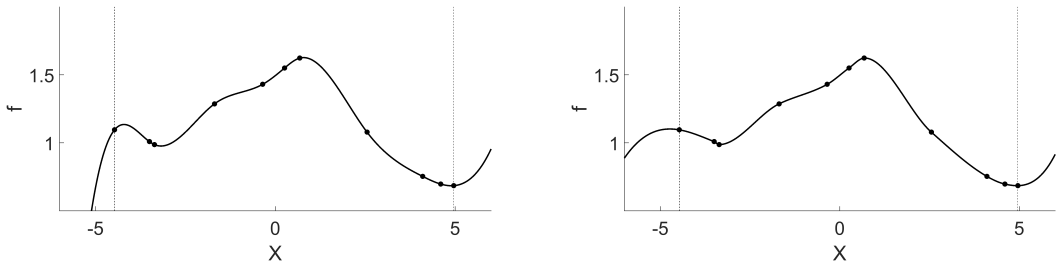


Abbildung 78: Interpolation mit einem kubischen Spline (links) und Interpolation mit einem kubischen Akima-Hermit-Spline (rechts); beide Beispiele wurden interpoliert mit Matlab

Neben der Interpolation über polynomiale Splines ist auch eine **Splineapproximation** möglich. Die Ergebnisse sind dann im Allgemeinen glatter als bei der Splineinterpolation, da nicht alle Datenpunkte exakt reproduziert werden müssen.

Interpolation mit radialen Basisfunktionen

Eine **radiale Basisfunktion** $\chi(\vec{x})$ ist eine glatte, stetig differenzierbare glockenförmige Funktion, welche an einem Basispunkt $\vec{x'}$ lokalisiert ist. Es gilt dabei

$$\chi_j(\vec{x}) = \chi_j(r_j) \text{ mit } r_j = d(\vec{x}, \vec{x_j'}).$$

r_j entspricht dem Abstand zwischen dem Punkt $\vec{(}x)$ und dem Basispunkt $\vec{(}x')$ der Basisfunktion χ. Es gibt viele verschiedene solcher radialer Basisfunktionen. Ein Beispiel ist die
Gaussian radial basis function: $\chi(r) = e^{-cr^2}, c > 0$.

Modellannahme: $\hat{f}$ lässt sich über eine Linearkombination aus k gegebenen radialen Basisfunktionen $\chi(r)_j$ mit $J = 1, \ldots, j, \ldots, k$ darstellen.

Ein interpolierter Wert $\hat{f}(\vec{x})$ an einer Position $\vec{x}$ kann über

$$\hat{f}(\vec{x}) = \sum_{j=1}^{k} a_j \chi_j(\vec{x})$$

bestimmt werden. Die Koeffizienten a_j werden wieder über die Lösung eines Gleichungssystems bestimmt.

Die folgenden Abbildungen zeigen das Interpolationsergebnis für das 1D Beispiel mittels *gaussian* radialen Basisfunktionen. Die Basispunkte entsprechen hier den gegebenen Datenpunkten, es wurden demzufolge $k = 11$ verschiedene Basisfunktionen verwendet.

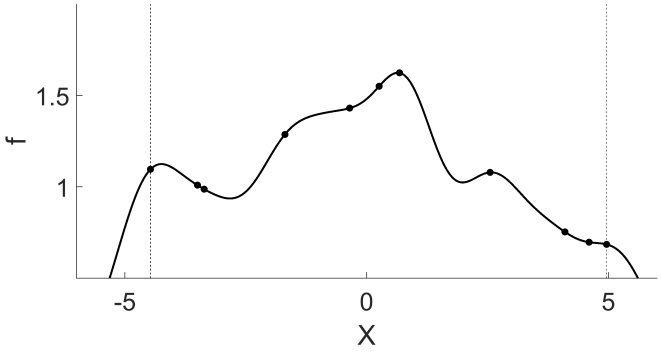


Abbildung 79: Interpolation mit 11 verschiedenen radialen Basisfunktionen vom Typ *gaussian*

Weitere Erläuterungen zum Thema Interpolation auf Punktmengen (gitterfrei) finden Sie in einem ScreenCast aus der Lehrveranstaltung "Einführung in die Geoinformatik".

6.3 Deterministische Interpolation über Vermaschungen

Verschiedene deterministische Interpolationsverfahren benötigen immer eine diskrete Unterteilung des zu interpolierenden Raums D. Im 1D ist dies eine Einteilung der Koordinatenachse in verschiedene Intervalle, im 2D sind es Vermaschungen (z.B. Voronoi-Vermaschung oder Delaunay Triangulierung) der Datenpunkte. Es handelt sich zumeist um **lokale** Interpolationen, bei denen die Nachbarschaft $N(\vec{x})$ über die Vermaschung bestimmt wird. Auch bestimmte Varianten der Spline-Interpolation verwenden Vermaschungen. Die im Folgenden vorgestellten Verfahren sind **exakte** Interpolationen.

Die grundsätzliche Ausgangssituation ist die Gleiche wie bei der Interpolation von verteilten Punktmengen. Für jeden Punkt $p_i \in P$, der ein Ausgangspunkt der Vermaschung darstellt, ist ein Wert f_i gegeben. Es gilt weiterhin $p_i = (\vec{x_i}, f_i) \rightarrow f_i = f_i(p_i) = f_i(\vec{x_i})$. Basierend auf den Werten $f_i(p_i)$ soll ein Wert $\hat{f}(\vec{x})$ an einer Position $\vec{x}$ vorhergesagt werden.

Die vorgestellten Verfahren werden an dem in der folgenden Abbildung gezeigten 2D Datensatz demonstriert.

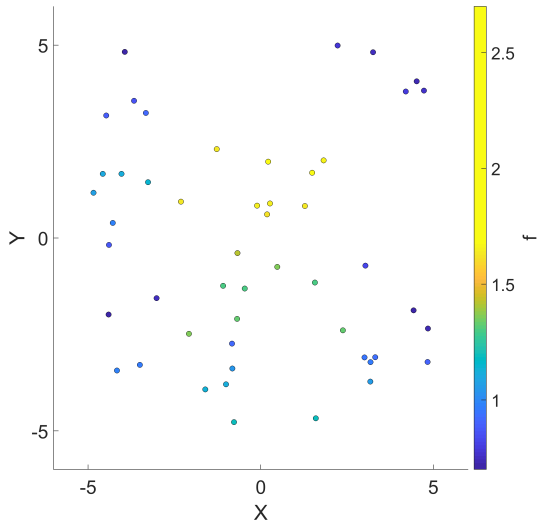


Abbildung 80: Beispieldaten zur Demonstration vermaschungsbasierter Interpolation

Interpolation auf Voronoi-Vermaschungen

Im Folgenden werden zwei Verfahren zur Interpolation vorgestellt, welche mit der Voronoi-Vermaschung der Datenpunkte in Beziehung stehen bzw. diese verwenden. Es handel sich einerseits um die **stückweise-konstante Interpolation**, auch als **Nearest-Neighbor-Interpolation** bezeichnet, und die Natural-Neighbor-Interpolation. Die Voronoi-Vermaschung der Beispieldaten ins in der nachfolgenden Abbildung dargestellt. Die gezeigten Interpolationsergebnisse basierend auf dieser Vermaschung.

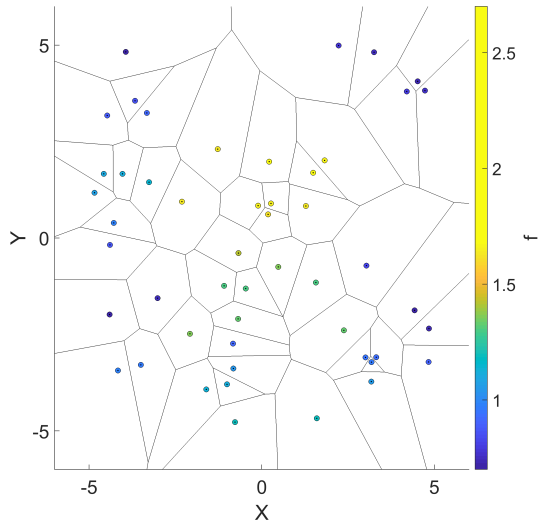


Abbildung 81: Voronoi-Vermaschung der Beispieldaten

Die Modellannahme der **stückweise-konstante Interpolation** ist, dass ein vorhergesagter Wert $\hat{f}(\vec{x})$ an einer Position $\vec{x}$ dem bekannten Wert $f_i(p_i)$ entspricht, der sich am äm nächsten" befindet. Dies lässt sich wie folgt formalisieren:

$$\hat{f}(\vec{x}) = \sum_{i=1}^{m} \mathbb{I}_{V_i}(\vec{x}) f_i.$$

Die Gewichte für jede Voronoi-Zelle $\mathbb{I}_{V_i}(\vec{x})$ sind gegeben über

$$\mathbb{I}_{V_i}(\vec{x}) = \begin{cases} 1 & d(\vec{x}, p_i) < d(\vec{x}, p_j) \quad \forall i \neq j \\ 0 & \exists j \neq i : d(\vec{x}, p_j) < d(\vec{x}, p_i) \end{cases}.$$

Dies bedeutet, dass immer nur ein Gewicht ungleich Null sein kann. Es handelt sich um das Gewicht für die Voronoi-Zelle, in der sich der Interpolationspunkt $\vec{x}$ befindet. Dies führt zu folgenden Eigenschaften:

- Innerhalb einer Voronoi-Zelle $V_i(p_i)$ ist $\hat{f}$ **konstant** und entspricht dem bekannten Wert $f_i(p_i)$.
- $\hat{f}$ weist Sprungstellen an den Zellgrenzen auf.

Theoretisch bedeutet dies, dass für einen Punkt $\vec{x}$ mit $d(\vec{x}, p_i) = d(\vec{x}, p_j)$ $\hat{f}(\vec{x})$ unbestimmt ist. In der Praxis wird dies meist ignoriert und es wird ein Wert $\hat{f}(\vec{x})$ festgelegt. Dieser ist dann aber abhängig von der jeweiligen Implementierung der Interpolation.

Die Gewichtsbestimmung basiert offensichtlich nur auf den Abständen zu den Zellzentren. In der Praxis bedeutet dies, dass die theoretisch zugrunde liegende Voronoi-Vermaschung für diese Interpolation nicht aufwändig aufgebaut werden muss. Dadurch ist es sehr einfach, die Nearest-Neighbor-Interpolation auf beliebige Dimensionen R^n zu erweitern, solange ein Entfernungsmaß $d(\vec{x}, p_i)$ definiert werden kann.

Die folgende Abbildung zeigt das Ergebnis der Nearest-Neighbor-Interpolation für den Beispieldatensatz. Die "scheinbar glatten" Parameterübergänge über den Zellgrenzen (schwarze Linien) sind ein Artefakt des Darstellungssystems.

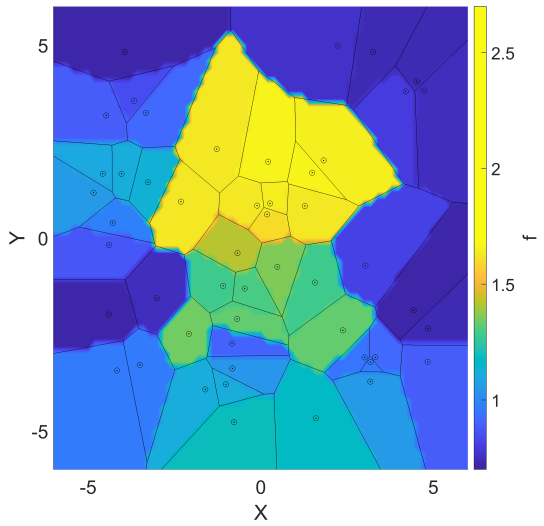


Abbildung 82: Nearest-Neighbor-Interpolation der Beispieldaten

Die ***Natural-Neighbor*-Interpolation** basiert, wie der Name schon andeutet, auf der Kenntnis der "natürlichen" (natural) Voronoi-Nachbarn eines zu interpolierenden Punktes $\vec{x}$. Die Modellannahme ist, dass $\hat{f}(\vec{x})$ sich über das gewichtete Mittel der bekannten Werte $f_i(p_{i_{\vec{x}}})$ an den natürlichen Nachbarn $p_{i_{\vec{x}}}$ von $\vec{x}$ ergibt.

Neben den Datenpunkten (p_i, f_i) muss die Voronoi-Vermaschung $\mathbf{V}(P)$ mit $p_i \in P$ bekannt sein. Für einen beliebigen Punkt $\vec{x}$ wird eine "sekundäre" Voronoi-Zelle $V(\vec{x})$ aufgebaut, welche seiner Voronoi-Zelle in der Voronoi-Vermaschung $\mathbf{V}(P \cup \{\vec{x}\})$ entspricht. Die Zellzentren $p_{i_{\vec{x}}}$ sind die natürlichen Nachbarn der Zelle $V(\vec{x})$, das heißt, ihre Zellen grenzen direkt an die Zelle $V(\vec{x})$. Das Polytop $V(p_{i_{\vec{x}}})$ ist das Schnittpolytop der Voronoizelle $V(\vec{x})$ und der Zelle $V(p_i) \in \mathbf{V}(P)$ mit $V(p_{i_{\vec{x}}}) = V(p_i) \cap V(\vec{x})$.

Dies wird in der folgenden Abbildung nochmals schematisch dargestellt. $\mathbf{V}(P)$ ist schwarz, die sekundäre Zelle $V(\vec{x})$ ist dunkel rot dargestellt. Das Schnittpolytop zur Zelle $V(p_3)$ ist in rot dargestellt.

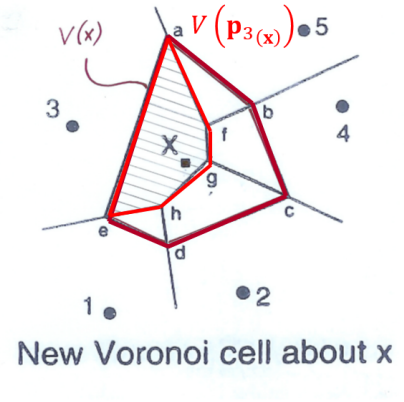


Abbildung 83: Schematischen Beispiel für ein Schnittpolytop $V(p_{i_{\vec{x}}})$

Ein so genanntes "natural neighbor" Gewicht $N_i(\vec{x})$ für den i-ten natürlichen Nachbarn von $\vec{x}$ kann nun wie folgt bestimmt werden:

$$N_i(\vec{x}) = \frac{m(V(p_{i_{\vec{x}}}))}{m(V(\vec{x}))}.$$

$m(V)$ ist dabei ein beliebiges Maß (*measure*) für ein Polytop, zum Beispiel die Länge eines Liniensegmentes in 1D, die Fläche eines Polygons oder das Volumen eines Polyeders. Das Gewicht ist also das Verhältnis des Maßes des Schnittpolytops zum Maß für das Polytop $V(\vec{x})$. Der interpoliert Wert an dieser Position $\vec{x}$ kann nun wie folgt bestimmt werden:

$$\hat{f}(\vec{x}) = \sum_{j=1}^{\text{Anzahl der nat. Nachbarn}} N_j(\vec{x}) f_j.$$

Der große Vorteil der Natural-Neighbor-Interpolation ist, dass die interpolierende Funktion $\hat{f}$ immer glatt ist (min. 1x steig differenzierbar) und auch bei Extrapolation vergleichsweise plausible Ergebnisse erzielt werden. Da für die Interpolationspunkte jedoch immer neue sekundäre Zellen bestimmt und diese Zellen verschnitten werden müssen, ist dieses Verfahren rechentechnisch vergleichsweise aufwändig. Es wird jedoch, aufgrund seiner gutartigen Eigenschaften, in der Praxis sehr häufig verwendet.

Die folgende Abbildung zeigt das Ergebnis der Natural-Neighbor-Interpolation für den Beispieldatensatz.

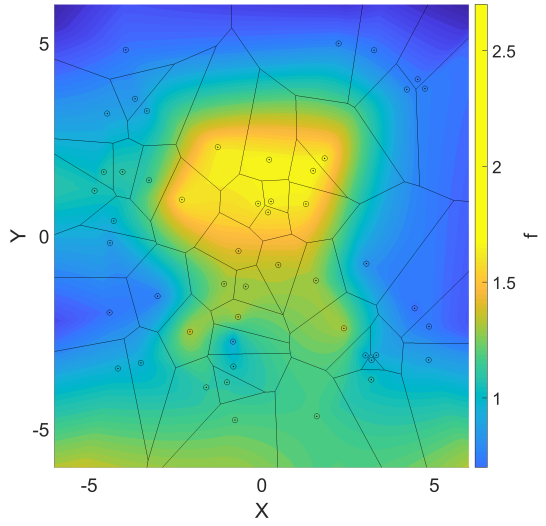


Abbildung 84: Nearest-Neighbor-Interpolation der Beispieldaten

Lineare Interpolation auf Dreiecken / Triangulierungen

Die Modellannahme, welche der linearen Interpolation auf einem Dreieck zugrunde liegt, ist, dass sich für einen Punkt innerhalb eines Dreiecks der interpolierte Wert aus dem gewichteten Mittel der bekannten Werte an den drei Eckpunkten ergibt.

Eine Grundvoraussetzung für eine solche Interpolation ist, dass eine Triangulierung $\mathbf{T}(P)$ der bekannten Datenpunkte (p_i, f_i) mit $p_i \in P$ vorliegt. Jedes Dreieck $T = \{p_i, p_j, p_k\} \in \mathbf{T}(P)$ verknüpft drei der bekannten Datenpunkte. Für den zu interpolierenden Beispieldatensatz ist die Delaunay Triangulierung in der folgenden Abbildung dargestellt. Die gezeigten Dreiecke überdecken die konvexe Hülle $[P]$ der Daten.

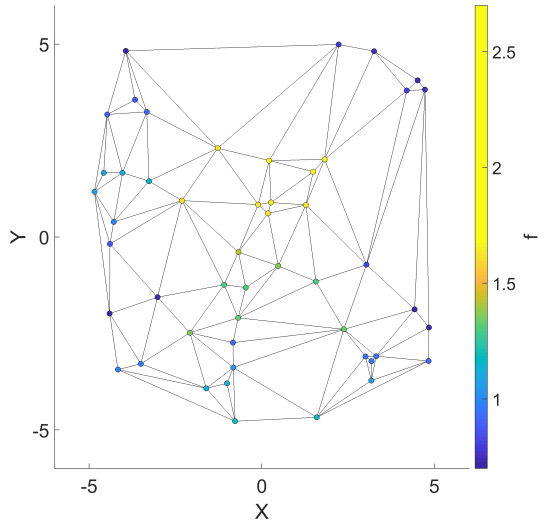


Abbildung 85: Delaunay Triangulierung der Beispieldaten

Im Allgemeinen lässt sich der zu interpolierender Wert $\hat{f}(\vec{x})$ wie folgt berechnen:

$$\hat{f}(\vec{x}) = \sum_{t=i,j,k} w_t f_t(p_t),$$

mit $\vec{x} \in T = \{p_i, p_j, p_k\}$. Für jeden der drei involvierten Datenwerte f_t mit $t = i, j, k$ muss ein Interpolationsgewicht w_t bestimmt werden. Dieses beruht auf dem Flächenverhältnis zwischen dem Dreieck T_t (Fläche A_t) und dem Dreieck T (Fläche A). Das Dreiecke T_t wird aus der dem Punkt p_t gegenüberliegenden Kante des Dreiecks T und dem Punkt $\vec{x}$ gebildet. Die Gewichte ergeben sich also aus $\{w_i = \frac{A_i}{A}, w_j = \frac{A_j}{A}, w_k = \frac{A_k}{A}\}$, wie die nachfolgenden Abbildung schematisch zeigt.

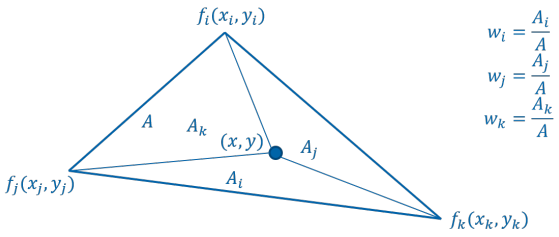


Abbildung 86: Notwendige Größen zur Berechnung der Gewicht für Interpolation eines Punktes innerhalb eines Dreiecks

Im 2D Fall mit $p_t = p_t(x_t, y_t)$ können die benötigten Größen wie folgt bestimmt werden:

$$A = \left| \det \begin{pmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ 1 & 1 & 1 \end{pmatrix} \right|, \; A_i = \left| \det \begin{pmatrix} x & x_j & x_k \\ y & y_j & y_k \\ 1 & 1 & 1 \end{pmatrix} \right|, \; A_j = \left| \det \begin{pmatrix} x_i & x & x_k \\ y_i & y & y_k \\ 1 & 1 & 1 \end{pmatrix} \right|,$$
$$A_k = \left| \det \begin{pmatrix} x_i & x_j & x \\ y_i & y_j & y \\ 1 & 1 & 1 \end{pmatrix} \right|.$$

Gemäß der *Cramerschen Regel* lässt sich dies zu $A = |(x_i - x_j)(y_j - y_k) + (x_j - x_k)(y_j - y_i)|$ umformen. Für die Flächen der Teildreiecke gilt dies analog.

Die Gewichte $w_T(\vec{x}) = \{w_i, w_j, w_k\}$ werden als die **baryzentrischen Koordinaten** des Punktes $\vec{x}$ bezüglich des Dreiecks T bezeichnet. Es gilt:

- Die Gewichte summieren immer zu 1: $\sum_{t=i,j,k} w_t = 1.$

- Die baryzentrische Koordinate des Punktes $p_i = p_i(x_i, y_i)$ lautet $w_T(x_i, y_i) = \{1, 0, 0\}$. Analoges gilt für die beiden anderen Punkte des Dreiecks.
- Ein Punkt mit $w_T(\vec{x}) = \{w_i = 0, w_j = \nu, w_k = 1 - \nu\}$ liegt auf der Kante zwischen den Punkten p_j und p_k. Die Kante wird durch diesem Punkt im Verhältnis $\frac{1-\nu}{\nu}$ geteilt. Analoges gilt für die beiden anderen Kanten des Dreiecks.

Die interpolierende Funktion $\hat{f}$ ist stück-weise linear und so nicht überall stetig differenzierbar. Über den Kanten der Dreiecke ist die erste Ableitung nicht stetig, innerhalb eines Dreiecks aber schon. Die interpolierende Funktion existiert nur über der Vermaschung, außerhalb davon ist sie nicht definiert. Extrapolation ist nativ nicht möglich. Dies ist gut in der nachfolgenden Abbildung zu erkennen, welche das Interpolationsergebnis für die Beispieldaten darstellt. Außerhalb der konvexen Hülle der Daten liegt kein Interpolationsergebnis vor.

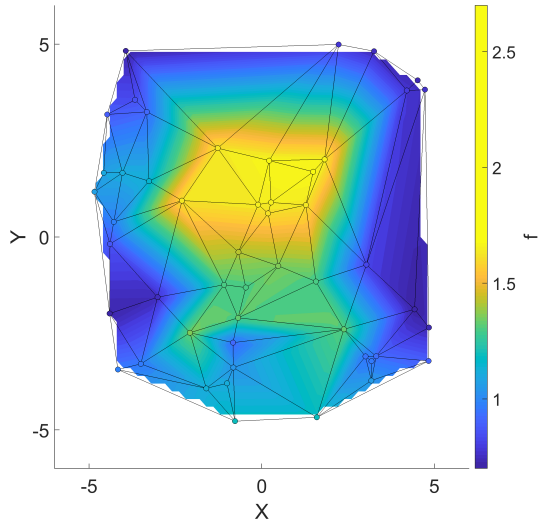


Abbildung 87: Lineare Interpolation auf der Triangulierung der Beispieldaten

Wenn eine lineare Interpolation auf einer Delaunay Triangulierung erfolgt, hat die interpolierende Funktion $\hat{f}$ noch eine zusätzliche Eigenschaft:

- $\hat{f}$ weist die **minimale Rauheit** im Vergleich zu einer linearen Interpolation auf allen anderen Nicht-Delaunay Triangulierungen der Datenpunkte auf.

Der Begriff *minimale Rauheit* bedeutet in diesem Fall, dass die zweite Ableitung über der gesamten Triangulierung minimiert wird. Die Parameterverteilung $\hat{f}$ ist maximal glatt.

Die Delaunay Triangulierung setzt damit die Geometrie/Topologie (Punktpositionen und ihre Triangulierung) des Sachverhalts automatisch mit einer Eigenschaft der zugehörigen linearen Interpolation in Beziehung. Dies ist bemerkenswert, da die Parameterwerte $f_i(p_i)$ beim Aufbau der Triangulierung nicht mit einbezogen / berücksichtigt wurden.

Weitere Erläuterungen zum Thema Interpolation auf Vermaschungen finden Sie in einem Screen-Cast aus der Lehrveranstaltung "Einführung in die Geoinformatik".

6.4 Geostatistik

Der folgende Abschnitt ist eine Einführung zum Thema Geostatistik und Kriging. Für eine tiefere Betrachtung des Themas existiert sehr viel spezialisierte Literatur, z.B. [Cre15]. An der TUBAF wird dies zum Beispiel im Kurs Multivariate Geostatistik behandelt.

Um mit der gängigen Geostatistikliteratur konsistent zu bleiben, wird im Folgenden eine geänderte Notation verwendet. Die bekannten Werte an dem Messpunkten werden als $z(\vec{x_i})$ mit $1 \leq i \leq m$ bezeichnet. $z^*(\vec{x})$ ist der so genannte **Schätzer** (*predictor* oder *estimator*) und entspricht $\hat{f}(\vec{x})$. Die Interpolationsgewichte werden im Folgenden mit λ bezeichnet.

Die Grundannahme der Geostatistik ist, dass die beobachten Werte z_i an den Beobachtungspositionen $\vec{x_i}$ Realisierungen einer Zufallsvariablen $Z(\vec{x_i})$ an diesen Punkten sind. Die Menge aller Zufallsvariablen $Z = Z(\vec{x}) \forall \vec{x}$ im Untersuchungsgebiet wird als *statistischer Prozess* oder *Zufallsfunktion* bezeichnet.

Das Ziel in der Geostatistik ist es, einen Schätzer $Z^*(\vec{x}) = \sum_{i=1}^{m} \lambda_i(\vec{x}) Z(\vec{x_i})$ für diese Zufallsfunktion zu finden. Die Gewichte sollen so bestimmt werden, dass $Z^*(\vec{x})$ ein guter Schätzer für $Z(\vec{x})$ ist. Um die Güte abschätzen zu können, ist es notwendig, dass dafür Kriterien definiert und auch gemessen werden können. Es soll also der *beste Schätzer* bezüglich eines gegebenen Kriteriums gefunden werden. Dafür ist es notwendig, die Zufallsvariablen und die Zufallsfunktion statistisch zu beschreiben.

Der Mittelwert der Zufallsfunktion $Z = Z(\vec{x}), \forall \vec{x} \in D$ entspricht dem Erwartungswert der Zufallsvariablen mit $m(\vec{x}) = \mathrm{E}Z(\vec{x})$ und wird als **Erwartungswertfunktion** (*expected value function*) bezeichnet. Die Varianz der Zufallsfunktion ist gegeben über $\mathrm{Var}Z(\vec{x}) = \mathrm{E}Z^2(\vec{x}) - m^2(\vec{x})$ und wird als **Varianzfunktion** bezeichnet (*variance function*). Die **zentrierte 2-Punkt Kovarianz** (*centered 2-point covariance*) ist die Kovarianz der beiden Zufallsvariablen $Z(\vec{x_1})$ und $Z(\vec{x_2})$ mit

$$\mathrm{Cov}\left(Z(\vec{x_1}), Z(\vec{x_2}))\right) = \mathrm{E}[Z(\vec{x_1}) - m(\vec{x_1})][Z(\vec{x_2}) - m(\vec{x_2})] = \mathrm{E}\left(Z(\vec{x_1})Z(\vec{x_2})\right) - m(\vec{x_1})m(\vec{x_2}).$$

Wenn sowohl $\mathrm{Var}Z(\vec{x})$ als auch $\mathrm{Cov}\left(Z(\vec{x_1}), Z(\vec{x_2}))\right)$ existieren, d.h. immer und überall endliche Werte annehmen, ist $Z(\vec{x})$ eine **Zufallsfunktion zweiter Ordnung** (*second-order random function*).

Das so genannte **2-Punkt Variogramm** (*two-point variogram*) ist definiert durch $2\gamma(\vec{x_1}, \vec{x_2}) = \mathrm{Var}\left(Z(\vec{x_1}) - Z(\vec{x_2})\right)$. Es gilt

$$2\gamma(\vec{x_1}, \vec{x_2}) = \mathrm{Var}\left(Z(\vec{x_1}) - Z(\vec{x_2})\right) = \mathrm{Var}\left(Z((\vec{x_1}) + \mathrm{Var}\left(Z((\vec{x_2}) - 2\mathrm{Cov}\left(Z((\vec{x_1}, Z(\vec{x_2})\right).\right.\right.\right.$$

Für $\mathrm{Var}\left(Z(\vec{x_1})\right) = \mathrm{Var}\left(Z(\vec{x_2})\right) =: C(0)$ und $\mathrm{Cov}\left(Z((\vec{x_1} - Z(\vec{x_2})\right) =: C(\vec{x_1}, \vec{x_2})$ gilt dann $\gamma(\vec{x_1}, \vec{x_2}) = C(0) - C(\vec{x_1}, \vec{x_2})$. $C(0)$ entspricht dem Maximalwert (*sill*) des 2-Punkt Variogramms.

Eigenschaften der Zufallsfunktion für die Geostatistik

Für eine geostatistische Interpolation stellt sich folgendes Problem: Als Anhaltspunkt für die gesuchte Zufallsfunktion existiert nur eine einzige Realisierung dieser Funktion, ausgewertet an einigen Punkten $\vec{x_i}$. Um überhaupt Aussagen über die zugrunde liegende Zufallsfunktion treffen zu können, muss angenommen werden, dass diese Zufallsfunktion bestimmte "gutartige" Eigenschaften aufweist, damit aus dieser einen Realisierung statistische Aussagen getroffen werden können.

Eine unbedingt notwendig Eigenschaft besteht in der Annahme, dass die so genannten Inkremente $z(\vec{x_1} + \vec{h}) - z(\vec{x_1})$ und $z(\vec{x_2} + \vec{h}) - z(\vec{x_2})$ ebenfalls Realisierungen einer eindeutigen Zufallsvariablen $\Delta(\vec{h}) := Z(\vec{x} + \vec{h}) - Z(\vec{x})$ sind, welche diese Inkremente unabhängig von den eigentlichen Positionen abbildet. Der Vektor $\vec{h}$ ist ein Verschiebungsvektor um eine Distanz h in eine spezifische Richtung.

Eine Zufallsfunktion ist **strikt stationär** (*strongly/strictly stationary*), wenn für ihre Momente (Erwartungswert, Varianz, Kovarianz), so sie denn existieren, gilt, dass sie invariant unter Verschiebungen sind mit

$$\mathrm{E}Z(\vec{x}) = m,$$

$$\mathrm{Cov}\left(Z(\vec{x}), Z(\vec{x} + \vec{h})\right) = \mathrm{E}[Z(\vec{x}) - m][Z(\vec{x} + \vec{h}) - m] = C(\vec{h}) \text{ und}$$

$$\mathrm{Var}\left(Z(\vec{x} + \vec{h}) - Z(\vec{x})\right) = \mathrm{E}[Z(\vec{x} + \vec{h}) - Z(\vec{x})]^2.$$

Demzufolge ist der Mittelwert überall konstant und die Kovarianz hängt nur von $\vec{h}$ ab, aber nicht von den beteiligten Positionen selbst. Ein alternativer Begriff in der Literatur für Stationarität / stationär ist Homogenität / homogen (*homogenity / homogeneous*).

Eine Zufallsfunktion wird als **schwach stationäre** (*weakly/second-order stationary*) Zufallsfunktion SRF bezeichnet, wenn nur noch gelten muss:

$$\mathrm{E}Z(\vec{x}) = m \text{ und}$$

$$\text{Cov}\left(Z(\vec{x}), Z(\vec{x}+\vec{h})\right) = \text{E}[Z(\vec{x}) - m][Z(\vec{x}+\vec{h}) - m] = C(\vec{h}).$$

Die Voraussetzungen an die Gutartigkeit der Varianz entfallen in diesem Fall.

Aus der Eigenschaft der (schwachen) Stationarität folgt, dass der Erwartungswert der Differenz der Zufallsfunktion an zwei Positionen Null ist:

$$\text{E}\left(Z(\vec{x}+\vec{h}) - Z(\vec{x})\right) = 0.$$

Es folgt weiterhin, dass die Varianz der Differenz der Zufallsfunktion an zwei Positionen dem doppelten Wert des 2-Punkt Variogramms des Abstands beider Positionen entspricht:

$$\text{Var}\left(Z(\vec{x}+\vec{h}) - Z(\vec{x})\right) = 2\gamma(\vec{h}) = 2(C(0) - C(\vec{h})).$$

Wenn die Richtung der Inkremente $\vec{h}$ für die Kovarianz keine Rolle spielt und die Kovarianz $C(h)$ so nur vom Abstand h abhängt, ist die SRF **isotrop**.

Eine Zufallsfunktion wird als **intrinsisch stationäre / intrinsische** Zufallsfunktion **IRF** bezeichnet, wenn die Zufallsfunktion ihrer Inkremente $\Delta(\vec{h})$ eine schwach stationäre Zufallsfunktion (SRF) ist. Dann gilt für alle $\vec{x}$ und jedes $\vec{h}$:

$$\text{E}\left(Z(\vec{x}+\vec{h}) - Z(\vec{x})\right) = \vec{a}^T\vec{h} \text{ und}$$

$$\text{Var}\left(Z(\vec{x}+\vec{h}) - Z(\vec{x})\right) = C(\vec{h}).$$

Für einen konstanten Vektor $\vec{a}$ wird der Erwartungswert der Inkremente der Zufallsfunktion als linearer Trend bezeichnet. Ist dieser lineare Trend Null mit $\text{E}\left(Z(\vec{x}+\vec{h}) - Z(\vec{x})\right) = \vec{a}^T\vec{h} = 0$, dann ist die Zufallsfunktion eine **stationäre IRF**.

Intrinsische Stationarität ist eine schwächere Anforderung an die Zufallsfunktion als schwache Stationarität. Diese ist wiederum eine schwächere Anforderung als strikte Stationarität. Wenn die Zufallsfunktion aber strikt stationär ist, ist sie automatisch auch schwach stationär und intrinsisch stationär.

Empirisches Variogramm und Variogrammmodell

Wenn die Zufallsfunktion schwach oder zumindest intrinsisch stationär ist, dann gilt für das Variogramm $\gamma(\vec{h}) = \frac{1}{2}\text{Var}\left(Z(\vec{x}+\vec{h}) - Z(\vec{x})\right)$. Dieses Variogramm ist ein wichtiger Bestandteil des Krigings, einer wichtigen Klasse von geostatistischen Interpolationsverfahren. Da die wahre Zufallsfunktion allerdings unbekannt ist, ist auch das wahre 2-Punktvariogramm unbekannt und muss basierend auf den bekannten Daten geschätzt / modelliert werden.

Die bekannten Daten $z(\vec{x_i})$ sind einzelne Realisierungen an diskreten Positionen einer isotropen SRF oder IRF. Für diese Zufallsfunktion kann dann das **empirische Variogramm** $\hat{\gamma}(h)$ wie folgt bestimmt werden:

$$\hat{\gamma}(h) = \frac{1}{2N_h}\sum_{i=1}^{N_h}\left(z(\vec{x_i}) - z(\vec{x_i}+h)\right)^2.$$

N_h ist die Anzahl aller Wertepaare, welche exakt den Abstand h aufweisen. **Das empirische Variogramm erlaubt es die räumliche Variabilität eines Datensatzes bezüglich des Punktabstandes abzuschätzen**.

In der Praxis wird der kontinuierliche Abstand h in eine Menge von K diskreten, äquidistanten Abstandsklassen $h = (h_1, \ldots, h_k, \ldots, h_K)$ unterteilt. N_{h_k} ist dann die Anzahl aller Wertepaare, deren Abstand in die Abstandsklasse h_k fällt mit $|\vec{x_i} - \vec{x_j}| \approx h_k$. Das empirische Variogramm lässt sich dann wie folgt abschätzen:

$$\hat{\gamma}(h_k) = \frac{1}{N_k}\sum_{|\vec{x_i}-\vec{x_j}|\approx h_k}\left(z(\vec{x_i}) - z(\vec{x_j})\right)^2; \forall h_k.$$

Häufig werden in in der Praxis in einem ersten Schritt alle Variogrammwerten zwischen Punktpaaren $\hat{\gamma}(\vec{x_i}, \vec{x_j}) = (z(\vec{x_i}) - z(\vec{x_j}))^2$ berechnet und dem Abstand $h(\vec{x_i}, \vec{x_j})$ der jeweiligen Punktpaares zugeordnet. Diese Wertepaare lassen sich dann in einer so genannten Variogrammwolke darstellen (siehe links in der nachfolgenden Abbildung). Diese Wertepaare lassen sich dann in die Abstandsklassen gruppieren. Im empirischen Variogramm wird dann für jede Abstandsklasse der Mittelwert aller Variogrammwerte $\hat{\gamma}(h_k)$ in dieser Klasse dem Abstand h_k zugeordnet. Ein Beispiel dafür ist rechts in der nachfolgenden Abbildung dargestellt.

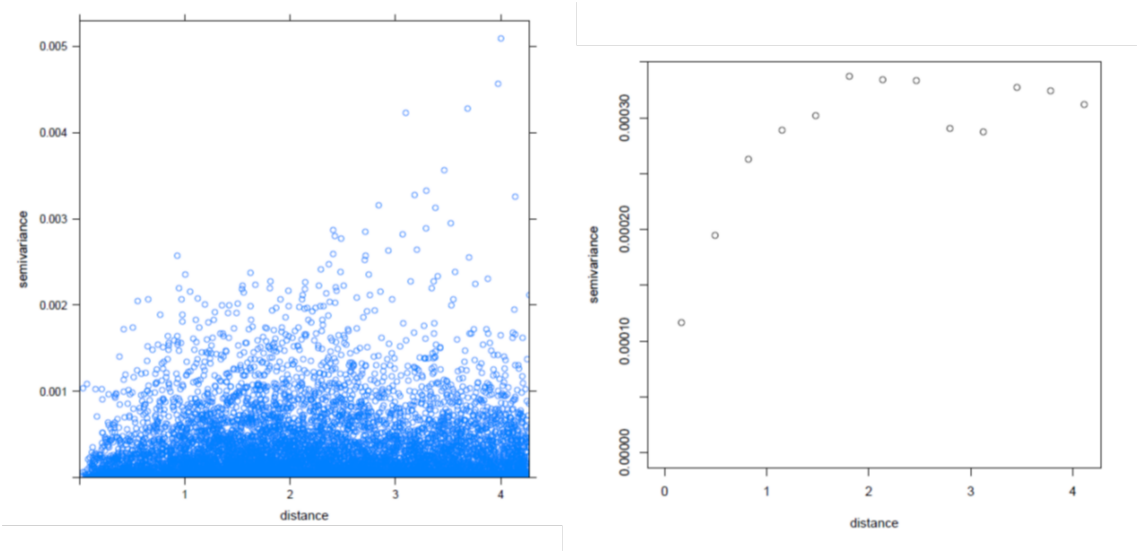


Abbildung 88: Variogrammwolke und zugehöriges empirisches Variogramm (Quelle: web.stanford.edu/class/stats253/lectures_2014)

Über das empirische Variogramm lässt sich eine analytische Funktion anpassen, welche das empirische Variogramm möglichst gut abbilden soll und als Modell für das theoretische 2-Punkt Variogramm dient. Diese analytische Funktion wird als analytisches Variogrammmodell bezeichnet.

Das analytische Variogramm dient als Modell für das theoretische 2-Punkt Variogramm $\gamma(h)$. Dies ist nur möglich, wenn die Zufallsfunktion wenigsten intrinsisch stationär, besser aber schwach stationär, ist. Nicht jede beliebige Funktion kann als Modell für ein 2-Punkt Variogramm dienen. Es gibt jedoch mehrere Typen von nutzbaren analytischen Funktionen, welche die notwendigen funktionalen Eigenschaften aufweisen. Mögliche Modelle für Variogramme sind zum Beispiel **exponentielle, sphärische oder gaussian Variogrammmodelle**, welche über die so genannten Variogrammparameter an das empirische Variogramm angepasst werden können. Die Variogrammparameter sind:

Sill: maximaler Variogrammwert,

Range: Distanz, ab der das Variogramm den maximalen Wert annimmt und der

Nugget (-effekt): Der Nugget beschreibt die Mikrovariabilität in einem Punkt. Er gibt an, in welchem Wert $\gamma_0 = \gamma(h = 0)$ die Variogrammfunktion die Y-Achse schneidet und repräsentiert die Varianz der Messwerte, wenn man in ein und derselben Lokation mehrfach messen würde.

Das analytische Variogrammmodell erlaubt es die Variogrammwerte $\gamma(h)$ für beliebige, kontinuierliche Abstände h zu bestimmen.

Im linken Teil der nachfolgenden Abbildung sind diese Parameter nochmals schematisch dargestellt. Im rechten Teil der nachfolgenden Abbildung sind verschiedene Variogrammmodelle dargestellt, welche ein gegebenes empirisch Variogramm repräsentieren. Die Wahl eines geeigneten Variogrammmodells und der passenden Parameter ist nicht nur abhängig von den verfügbaren bekannten Messdaten, sondern auch von der Natur"des zugrundeliegen Sachverhalts.

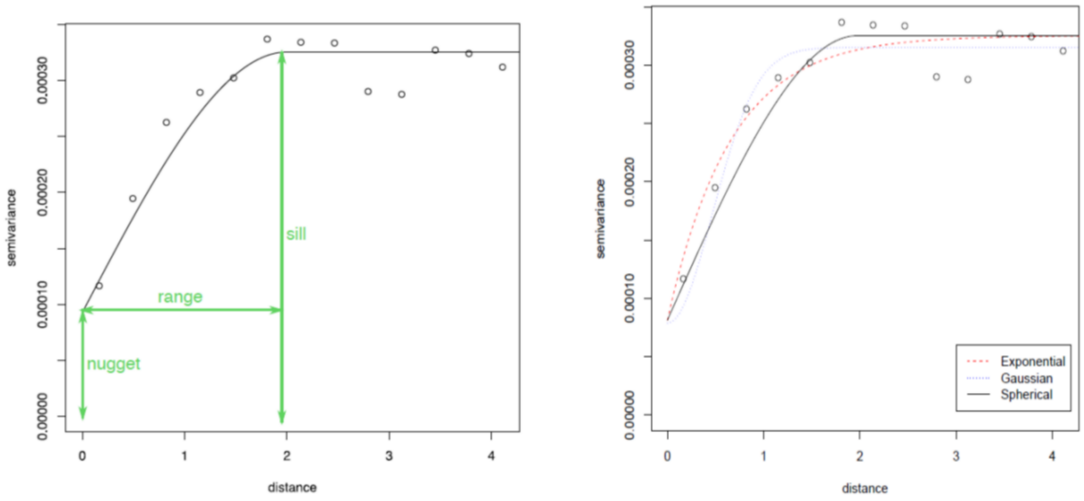


Abbildung 89: Variogrammparameter und Variogrammmodelle bzgl. eines empirischen Variogramms (Quelle: web.stanford.edu/class/stats253/lectures_2014)

Kriging

Kriging ist eine weit verbreitete Gruppe von geostatistisches Verfahren zur räumlichen Vorhersage. Wie in jedem klassischen linearen Interpolationsproblem soll eine Vorhersage $z^*(\vec{x})$ an einer unbeprobten Lokation basierend auf einer Linearkombination bekannter Werte $z(\vec{x_i})$ an beprobten Lokationen in der Form

$$z^*(\vec{x}) = \sum_{i=1}^{m} \lambda_i(\vec{x}) z(\vec{x_i}) + \lambda_0(\vec{x})$$

erstellt werden. Die Gewichte λ sollen auf der räumlichen Korrelation der Daten basierend.

Es soll ein Schätzer Z^* für die unbekannte Zufallsfunktion Z gefunden werden, welche den Daten zugrunde liegt. Diese wird für die folgenden Betrachtungen als **isotrop** angenommen. Bei Kriging ist es das Ziel, den **besten linearen unverzerrten Schätzer** (***best linear unbiased estimator - BLUE***) zu bestimmen. Unverzerrt (*unbiased*) bedeutet hier, dass der Erwartungswert des Schätzers dem Erwartungswerts der zugrunde liegenden Zufallsfunktion entspricht und somit der Erwartungswert der Differenz zwischen einem geschätzten Wert an einer Position und dem "wahren" Wert an dieser Position Null ist:

$$\mathrm{E}[Z^*(\vec{x})] = \mathrm{E}[Z(\vec{x})] \rightarrow \mathrm{E}[Z^*(\vec{x}) - Z(\vec{x})] = 0.$$

Des weiteren soll die Varianz der Abweichung des Schätzers von der unbekannten Zufallsfunktion minimal sein mit

$$\mathrm{Var}[Z^* - Z] \rightarrow \min..$$

Die Varianz wird über den Ausdruck

$$\frac{\partial}{\partial \lambda_i} \mathrm{Var}[Z^* - Z_i] = 2 \sum_{j=1}^{m} (\lambda_j \mathrm{Cov}[Z_j, Z_i]) - 2\mathrm{Cov}[Z_i, Z]$$

minimiert. Die dafür benötigten Kovarianzen $\mathrm{Cov}[Z_j, Z_i] = C(h_{ij})$ und $\mathrm{Cov}[Z_i, Z] = C(h_{i0})$ sind allerdings unbekannt. Ohne weitere Annahmen an Z kann Kriging nicht durchgeführt werden. Ist Z jedoch mindestens eine intrinsische Zufallsfunktion (IRF), lassen sich diese Kovarianzen aufgrund der Beziehung $\gamma(h_{ij}) = C(0) - C(h_{ij}) \rightarrow C(h_{ij}) = C(0) - \gamma(h_{ij})$ durch die über ein bekanntes Variogrammmodell ermittelten Werte $\gamma(h_{ij})$ ersetzen. h_{ij} ist dabei der Abstand zwischen den beiden Datenpunkten $\vec{x_i}$ und $\vec{x_j}$, h_{j0} ist der Abstand zwischen dem j-ten Datenpunkt und dem Interpolationspunkt $\vec{x}$ und $C(0)$ ist die maximale Kovarianz (= Sill des Variogramms). Die Richtung von $\vec{x_i}$ nach $\vec{x_j}$ braucht hier nicht berücksichtigt zu werden. Aufgrund der Isotropie der

Zufallsfunktion spielt nur der Abstand zwischen zwei Punkten eine Rolle und nicht, wie diese Punkt relativ zueinander liegen.

Damit Kriging angewandt werden kann, muss die so genannte **Stationaritätsannahme** erfüllt sein. Diese besagt, dass der **zugrunde liegende Zufallsprozess stationär ist** oder alternativ, dass die **zugrunde liegende Zufallsfunktion eine stationäre Zufallsfunktion (mindestens eine IRF) ist**. Ist diese Annahme nicht erfüllt, kann Kriging nicht zuverlässig durchgeführt werden!

Für die verschiedenen Kriging-Ansätz wird zwischen verschiedenen weiteren Ausgangsannahmen unterschieden:

Simple Kriging (SK): der Erwartungswert der Zufallsfunktion ist bekannt und konstant $\mu = \mathrm{E}[Z]$;

Ordinary Kriging (OK): der Erwartungswert ist konstant, aber unbekannt;

Universal Kriging (UK): der Erwartungswert selbst ist unbekannt und nicht konstant, kann aber über eine funktionale Abhängigkeit abgeschätzt werden mit $\mu(\vec{x}) = \mathrm{E}[Z(\vec{x})]$.

Im Fall von Simple oder Ordinary Kriging wird Z typischerweise als SRF angenommen, im Fall von Universal Kriging muss Z nur eine IRF sein. Aus den oben gezeigten Überlegungen zur Minimierung der Varianzen ergeben sich für ein Simple Kriging folgende Gleichungen

$$\sum_{i=1}^{m} \lambda_i (C(0) - \gamma(h_{ij})) = C(0) - \gamma(h_{j0}).$$

Zur Bestimmung der Gewichte λ_i muss also das Gleichungssystem $K_{SK}\lambda_{SK}(\vec{x}) = b_{SK}(\vec{x})$ gelöst werden mit

$$K_{SK} = \begin{bmatrix} C(0)-\gamma(0) & C(0)-\gamma(h_{21}) & \cdots & C(0)-\gamma(h_{m1}) \\ \vdots & \vdots & \ddots & \vdots \\ C(0)-\gamma(h_{1m}) & C(0)-\gamma(h_{2m}) & \cdots & C(0)-\gamma(0) \end{bmatrix}, \lambda_{SK} = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_m \end{bmatrix} \text{ und}$$

$$b_{SK} = \begin{bmatrix} C(0)-\gamma(h_{10}) \\ \vdots \\ C(0)-\gamma(h_{m0}) \end{bmatrix}.$$

Neben der Vorhersage eines Wertes $z^*(\vec{x})$ kann über Kriging zusätzlich eine so genannte **Kriging-varianz** σ^2_{SK} für den Interpolationspunkt $\vec{x}$ angeben werden. Diese ergibt sich über $\sigma^2_{SK}(\vec{x}) = \lambda_{SK}(\vec{x})^T \cdot b_{SK}(\vec{x})$ und kann als Maß für die Unsicherheit der Vorhersage am Punkt $\vec{x}$ angesehen werden. Analoge Beziehungen existieren auch für die beiden anderen Kriging-Ansätze.

Wenn man von einer SRF oder IRF ausgeht, welche den Daten zugrunde liegt, und dadurch die Stationaritätsanforderung erfüllt ist, ist Kriging eine exakte Interpolationsmethode. Der Schätzer Z^* ist zudem der *beste lineare unverzerrte Schätzer (BLUE)* für Z und basiert auf der räumlichen Kovarianz bzw. dem Variogramm der gegebenen bekannten Daten. Neben einem Schätzer für den Wert an einer unbeprobten Lokation kann über Kriging zusätzlich die Krigingvarianz für diese Lokation als Maß für die Unsicherheit der Schätzung bestimmt werden.

Kriging-Workflow:

1. **Gegeben:** Werte $z(\vec{x_i})$ als Realisierungen einer Zufallsfunktion Z.
   Gesuch: Bester Schätzer (BLUE) Z^* für die Zufallsfunktion, zur Vorhersage möglicher Werte $z^*(\vec{x})$ an unbeprobten Lokationen.
2. **Stationaritätsannahme ist erfüllt!** Z ist wenigstens eine IRF, besser eine SRF.
3. Berechnung des empirischen Variogramms $\hat{\gamma}$ für die Daten und Modellierung des zugehörigen analytischen Variogrammmodells γ.
4. Aufbau der Kriging-Matrix K und der rechten Seite $b(\vec{x})$ basierend auf γ.
5. Berechnung der Gewichte $\lambda_i(\vec{x})$ durch Lösen des Gleichungssystems $\lambda_i(\vec{x}) = K^{-1}b(\vec{x})$.
6. Vorhersage von $z^*(\vec{x})$ mit $z^*(\vec{x}) = \sum_{i=1}^{m} \lambda_i(\vec{x}) z(\vec{x_i})$.

Weitere Erläuterungen zum Thema Geostatistik / Kriging finden Sie in einem ScreenCast aus der Lehrveranstaltung Einführung in die Geoinformatik und im OPAL-Kurs Kurs Multivariate Geostatistik.

7 Räumliche Transformationen

Jede GIS-Operation kann als eine Transformation angesehen werden ([BC94]). Neben den bereits beschrieben wichtigen Transformationen zu Dateneingabe (siehe Abschnitte zu *Koordinatentransformation* und *Datenmodellen*) sowie der allgemeinen Übertragung und Vorhersage von Attributwerten (siehe Abschnitt zur *Interpolation*) befassen sich weitere Klassen von Transformationen mit Änderungen zwischen und innerhalb von prinzipiellen Objekttypen (Punkte, Linien, Polygone). Dabei sind zum Beispiel so genannte **Punkt-zu-Fläche-Transformationen** (point-to-area-transformation, [BC94]) sehr weit verbreitet. Diese Klasse von Transformationen fasst sich mit dem Problem, wie sich Punktinformationen in der Fläche auswerten lassen. Dies lässt sich grundsätzlich über Interpolation lösen. Im Folgenden wird aber zusätzlich näher auf nicht-interpolierende Ansätze zur Lösung dieses Problems eingegangen. Eine weitere Klasse befasst sich mit der Änderung und Analyse der räumlichen Ausdehnung / Form räumlicher Objekte (***dilation transformations***). Dabei ist vor allem die Transformation **Dilation / Buffering** wichtig, um Objekte "virtuell" in ihrer Fläche zu vergrößern, um weitergehende räumliche Aussagen treffen zu können. Für Rasterdaten umfasst diese Klasse zudem Transformationen wie **Filterung** (kontinuierliche Daten) und **morphologische Operatoren** (diskrete Daten). Eine weitere Klasse von Transformationen (***sampling transformations***) befasst sich mit der Auswertung von Objekten an Punktpositionen (*sampling*) und der Änderung der Objektauflösung (*resampling*).

7.1 Punkt-zu-Fläche-Transformationen

Punkobjekte sind 0-dimensional Objekte und haben demzufolge zwar eine Position, aber keine Ausdehnung. Um sie aber in einem 2D GIS oder auf einer Karte abbilden und auswerten zu können, muss die Punktinformation (Lage und Attribute) in die Fläche transformiert werden.

Abbildung der Punktdichte (*density mapping*)

Beim *density mapping* ist das Ziel, die Punktdichte flächenhaft abzubilden. Die Punktdichte ρ_P entspricht dabei der Anzahl an Punkten pro Flächeneinheit ([BC94]). Dabei wir die Fläche in eine Menge nicht überlappender Flächenelemente (unregelmäßiger Polygone oder Pixel) zerlegt. Sind die Flächenelemente Polygone, ist das resultierende Objekt eine Vermaschung, werden Pixel verwendet, ist es ein Rasterobjekt.

Gegeben ist eine Menge P mit m unterschiedlichen Punkt $\vec{x_i}, i = 1, \ldots, m$ und eine Zerlegung $\mathbf{F}$ der Fläche. Für jedes Flächenelement $F \in \mathbf{F}$ wird gezählt, wie viele Punkte in diesem Flächenelement liegen. Diese Anzahl wird durch die Fläche des Elements geteilt. der formale Ausdruck lautet wie folgt:

$$\rho_P(F) = \frac{\sum_{i=1}^{m} \mathbb{I}_F(\vec{x_i})}{A_F} \text{ mit } \mathbb{I}_F(\vec{x_i}) = \begin{cases} 1 & \vec{x_i} \text{ liegt in } F \\ 0 & \vec{x_i} \text{ liegt nicht in } F \end{cases}.$$

A_F ist die Fläche des Flächenelements. Jedem Flächenelement wird dabei ein Wert für Punktdichte mit $\rho_P(F) \to F$ zugeordnet.

Density mapping ist damit eine flächenhafte Repräsentation verteilter Punktobjekte, welche für die Visualisierung und Auswertung verwendet werden kann. Es wird nur die Lage der Punkte berücksichtigt, an den Punktobjekten vorliegende Attribute / Variablen / Parameter werden nicht mit einbezogen. Bei sehr unregelmäßig verteilten Punkten kann es passieren, dass sich die Punktdichte zwischen zwei benachbarten Flächenelementen stark ändert. Dies kann zum Beispiel über einen so genannten *moving-window* Ansatz abgemildert werden. Für die Berechnung der Punktdichte werden dabei nicht nur die Punkte in einem Flächenelement gezählt sondern zusätzlich auch die in den Nachbarelementen. Dieser Wert wird durch die Fläche des Elements und aller seiner Nachbarn dividiert.

Transformation von Punktinformationen in Flächenobjekte

Methoden zur Transformation von Punktinformation in flächenhafte Objekte werden in interpolative und nicht-interpolative Verfahren unterteilt ([BC94]). Interpolative Verfahren verwenden Interpolation, um die initial an verteilten Punkten vorliegenden Daten in der Fläche zu visualisieren und auszuwerten. Bei nicht-interpolativen Verfahren werden direkt die an einer Punktlokation

vorliegenden Daten (ein oder mehrere Attribute) einem Flächenobjekt zugeordnet. Ziel ist hier in erster Linie eine 1-1-Zuordnung zwischen Punkten und z.B. Polygonen, das heißt jedem Punkt wird ein separates Polygon zugeordnet. Dabei wird die Attributtabelle der Punkte direkt in die Attributtabelle der Polygone überführt. Manche Verfahren ermöglichen auch eine n-1-Zuordnung, das heißt die Werte an mehreren Punkten werden einem Polygon zugeordnet. Dafür muss der Attributwert eines Polygons erst aus den Attributwerten der Punkt berechnet werden (z.B. Mittelwert, Median, ...).

Gitterverfahren: Die von den Punktdaten überdeckte Fläche wird in ein regelmäßiges Gitter zerlegt. Der Attributwert für jede Gitterzelle wird aus den Attributwerten der Punkte innerhalb der Zelle berechnet. Liegen keine Punkte in der Zelle, wird ein *default*-Wert (z.B. **NULL**, **NaN**=*not a number*, **NA**=*not available*, ...) der Zelle zugeordnet.

Einflusszonen: Jeder Punkt wird einem Polygon zugeordnet. Dieses Polygon definiert die Einflusszone des Punktes und erhält den Attributwert des in ihm liegenden Punktes. Häufig werden hierfür kreisförmige Polygone mit gegebenem Radius verwendet, das Zentrum ist der zugehörige Datenpunkt. Sich überlappende Polygone werden zu einem Polygon vereinigt und der Attributwert muss berechnet werden. Regionen, welche nicht von einem Polygon überdeckt sind, erhalten einen *default*-Wert.

Voronoi-Polygone: Basierend auf den gegebenen Punkten wird eine Voronoi-Vermaschung durchgeführt, jeder Voronoi-Zelle wird der Wert ihres zugrunde liegenden Punktes zugeordnet. Das Ergebnis ist eine *Nearest-Neighbor*-Interpolation. Dieses Verfahren wird nach Bonham-Carter (1994,[BC94]) den nicht-interpolativen Verfahren zugeordnet, da laut Bonham-Carter (1994,[BC94]) interpolative Verfahren immer ein kontinuierliches Interpolationsergebnis (d.h. keine Parametersprünge) aufweisen müssen. Diese Bedingung wird aber in der Praxis nicht generell an Interpolation gestellt.

Die folgende Abbildung zeigt Beispiele für die flächenhafte Repräsentation von Punktinformationen. Neben den oben erläuterten Verfahren sind noch weitere Verfahren möglich, auf die hier nicht näher eingegangen wird. Ein Beispiel wäre **F** in der folgenden Abbildung, so genannte *catchment basins*, welche ähnlich zu einer Voronoi-Vermaschung den Punktdaten Polygone zuordnen.

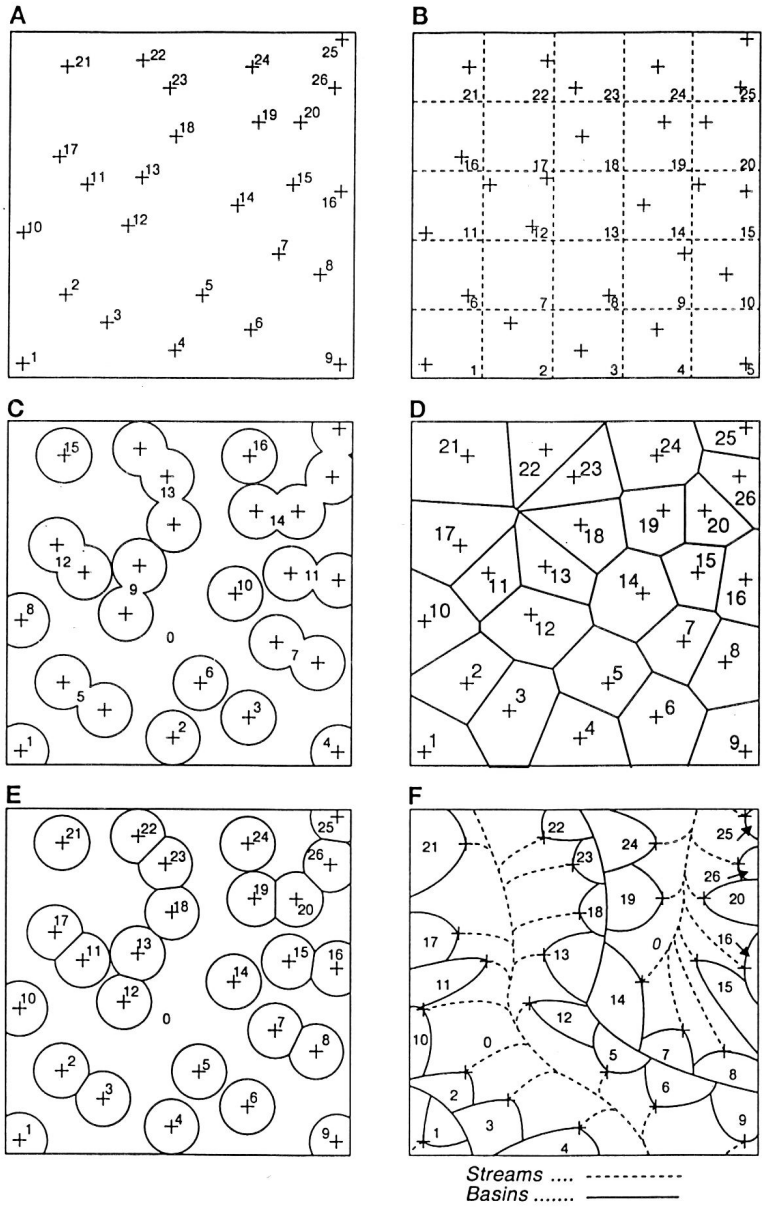


Abbildung 90: Beispiele für verschiedenen Transformationen von Punktinformationen in Flächenobjekte. **A**: Gegebenen Punktinformationen, **B**: Gitterverfahren, **C**: Einflusszonen, **D**: Voronoi-Polygone, **E**: Vornoi-Polygone, repräsentiert über Kreispolygone, **F**: catchment basins ([BC94])

7.2 *Sampling* Transformationen

Ein häufiges Problem in GIS ist, dass Linien- und Flächenobjekte punktweise ausgewertet werden sollen. Das heißt, es sollen zum Beispiel die Werte einer Attributtabelle für Flächenobjekte in eine Attributtabelle für Punkte übertragen werden. Auch die gemeinsame Auswertung von überlagerten Flächenobjekten zählten zu diesen Problemen. Eine Sonderform ist das so genannte *Resampling*, also die Veränderung der Auflösung von Objekten durch Veränderung der Anzahl der zugehörigen Stützpunkte. Zusätzliche Stützpunkte lassen sich zumeist leicht hinzufügen. Soll allerdings die Anzahl der Stützpunkte reduziert werden, stellt sich die Frage, welche Stützpunkte entfernt werden "dürfen", ohne zu viel Information zu verlieren.

Fläche-zu-Punkt-Transformationen

Beim Punktsampling von Flächenobjekten soll für eine Punktlokation $\vec{x}$ ein Attributwert basierend auf den Attributwerten von Flächenobjekten ermittelt werden. Der Punktlokation wird dabei genau der Attributwert des Flächenobjektes zugeordnet, in dem sich die Punktlokation befindet. Dafür muss dieses zugehörige Flächenobjekt (Polygon) gefunden werden. Im Allgemeinen wird diese Klasse von Verfahren als "Punkt-in-Polygon"-Tests bezeichnet. Es existieren viele verschiedenen Methoden, um solche Tests durchzuführen, abhängig von der Anwendung und den gegebenen Daten. Im folgenden werden drei Ansätze unterschiedlicher Komplexität vorgestellt:

***Bounding-Box*-Test / Punkt-in-Rechteck-Test:** Mit dem sogenannten *Bounding-Box*-Test kann sehr einfach und effizient überprüft werden, ob sich ein Punkt $\vec{x} = (x, y)$ in einem achsparallelen Rechteck befindet. Ein achsparalleles Rechteck zeichnet sich dadurch aus, dass seine Kanten immer parallel zu einer der Koordinatenachsen verlaufen. In diesem Fall lässt sich ein solches Rechteck R nur basierend auf den minimalen und maximalen Koordinatenwerten seiner Eckpunkte beschreiben mit $R = (x_{min}, x_{max}, y_{min}, y_{max})$. Um im 2D zu ermitteln, ob sich $\vec{x}$ innerhalb von R befindet, müssen nur zwei Bedingungen erfüllt sein:

$$\vec{x} \in R \text{ wenn gilt } \begin{cases} x_{min} \geq x \leq x_{max} \\ \text{und} \\ y_{min} \geq y \leq y_{max} \end{cases} .$$

Das heißt, der x-Wert des Punktes muss innerhalb des X-Achsenabschnittes liegen, der auch vom Rechteck überdeckt wird **und** der y-Wert des Punktes muss innerhalb des Y-Achsenabschnittes liegen, der auch vom Rechteck überdeckt wird.

In der folgenden Abbildung ist dies am Beispiel zweier Punkte $a_1 = (x_1, y_1)$ und $a_2 = (x_2, y_2)$ und des grauen Rechtecks $R = (x_{min}, x_{max}, y_{min}, y_{max})$ dargestellt. $a_1 = (x_1, y_1)$ liegt definitiv innerhalb des Rechtecks, da beide Bedingungen erfüllt sind: $\vec{x} \in R \rightarrow \begin{cases} x_{min} \geq x_1 \leq x_{max} \\ y_{min} \geq y_1 \leq y_{max} \end{cases}$. $a_2 = (x_2, y_2)$ liegt demzufolge nicht innerhalb von R, da mindestens eine Bedingung mit $x_2 < x_{min}$ nicht erfüllt ist.

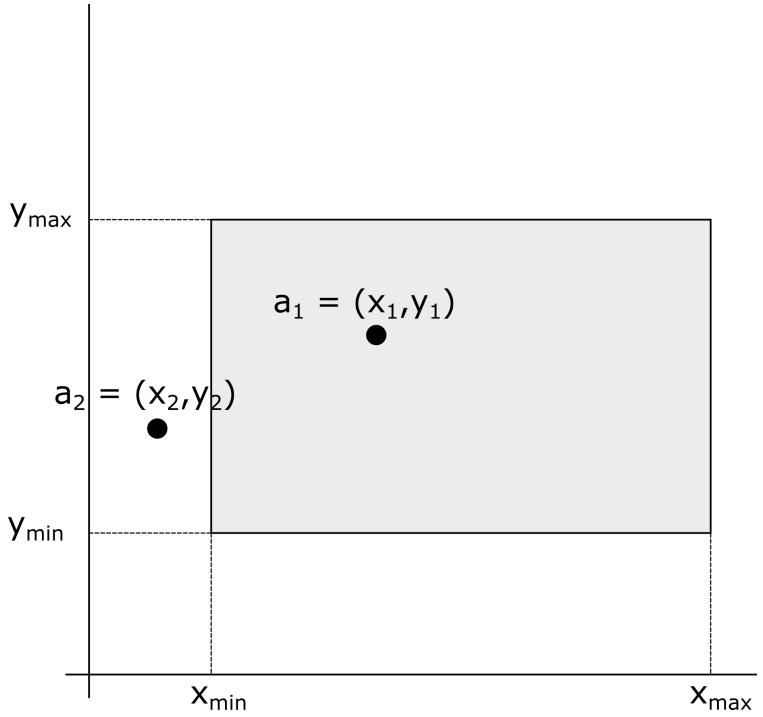


Abbildung 91: Schematische Darstellung einer Situation für einen "Punkt-in-Rechteck"-Test

Dieses Testverfahren lässt sich sehr leicht auch auf höhere Dimensionen erweitern und kann **immer** sehr effizient durchgeführt werden. In der Praxis treten achsparallel Rechtecke als Geoobjekte, abgesehen von Rasterdaten / Pixeln, allerdings nur sehr selten auf. Dieser Test wird aber genutzt, um geeignete Kandidaten für aufwändigere "Punkt-in-Polygon"-Test zu ermitteln. Jedes beliebige Polygon lässt sich über ein achsparalleles Rechteck repräsentieren, wenn die minimalen und maximalen x- und y-Werte seiner Eckpunkte bekannt sind. Dieses Rechteck wird dann als ***axis-aligned bounding-box*** (*AABB*, achsparalleles Begrenzungsrechteck) des zugehörigen Polygons bezeichnet. Ein Punkt, welcher bereits außerhalb dieser *Bounding-Box* liegt, kann in keinem Fall im zugehörigen Polygon liegen, unabhängig von dessen tatsächlicher Form. Ein aufwändigerer "Punkt-in-Polygon"-Test zwischen dem Punkt und diesem Polygon ist somit nicht mehr notwendig.

Punkt-in-Dreieck-Test: Dreiecke treten sehr häufig in verschiedenen Anwendungen auf, zudem existieren mit Triangulierungen spezielle Vermaschungen, bei denen bekannt ist, dass sie nur aus Dreiecken bestehen. Deshalb werden hier "Punkt-in-Polygon"-Tests verwendet, welche auf Dreiecke optimiert sind.

Ein häufiger "Punkt-in-Dreieck"-Test basiert auf der Tatsache, dass eine Gerade den Raum R^2 in zwei Halbräume teilt. Ein Dreieck ergibt sich aus dem Schnitt der drei Halbräume, welche sich jeweils links (oder rechts je nach Laufrichtung) der Geraden befinden, welche sich durch die Kanten des Dreiecks ergeben. Ein Punkt, welcher sich in allen drei Halbräumen befindet (bzw. in der Schnittmenge der drei Halbräume), befindet sich automatisch im Inneren des Dreiecks.

Für ein Dreieck $\Delta(\vec{p_i}, \vec{p_j}, \vec{p_k})$ ist immer bekannt, dass der einer Kante $(\vec{p_i}, \vec{p_j})$ gegenüberliegende Punkt $\vec{p_k}$ im Dreieck liegen muss. Ein Punkt $\vec{x}$, der im gleichen Halbraum wie $\vec{p_k}$ bezüglich der Kante $(\vec{p_i}, \vec{p_j})$ liegt, kann also potentiell innerhalb des Dreiecks liegen. Ob $\vec{x}$ und $\vec{p_k}$ im gleichen Halbraum liegen, kann über

$$\omega_{ij}(\vec{x}) = ((\vec{p_j} - \vec{p_i}) \times (\vec{p_k} - \vec{p_i})) \cdot ((\vec{p_j} - \vec{p_i}) \times (\vec{x} - \vec{p_i}))$$

bestimmt werden. Für $\omega_{ij}(\vec{x}) > 0$ liegen beide Punkt im gleichen Halbraum. $\vec{x}$ liegt also innerhalb von Δ, wenn gilt:

$$\omega_{ij}(\vec{x}) > 0 \text{ und } \omega_{jk}(\vec{x}) > 0 \text{ und } \omega_{ki}(\vec{x}) > 0.$$

Ist bereits eine dieser Bedingungen nicht erfüllt, kann $\vec{x}$ nicht innerhalb des Dreiecks liegen.
In der nachfolgenden schematischen Abbildung liegt der Punkt P_3 der Kante (P_1, P_2) gegenüber. Der Punkt a_1 liegt im gleichen Halbraum bezüglich dieser Kante wie Punkt P_3 und kann deshalb potentiell im Inneren des Dreiecks liegen. Ob er tatsächlich in Inneren liegt, muss noch bezüglich der anderen beiden Kanten getestet werden. Punkt a_2 liegt nicht im gleichen Halbraum bezüglich der Kante wie Punkt P_3 und kann deshalb nicht im Inneren des Dreiecks liegen. Ein Test bezüglich der anderen Kanten ist nicht mehr notwendig.

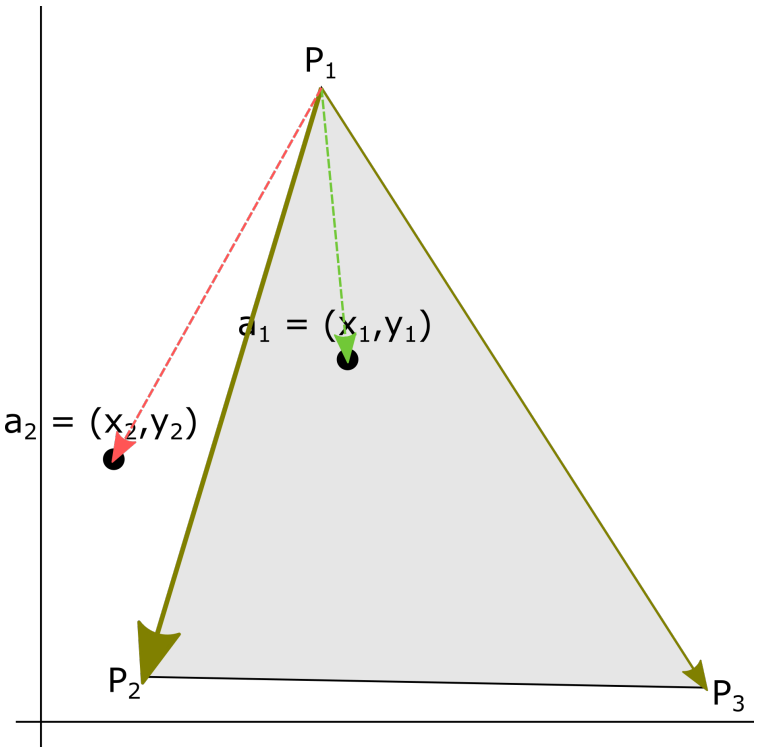


Abbildung 92: Schematische Abbildung für die Situation eines "Punkt-in-Dreieck"-Tests

Dieser Test ist für jede Kante vergleichsweise aufwändig zu berechnen (2 Kreuzprodukte und ein Skalarprodukt). Der Vorteil ist aber, dass sobald der Test für eine Kante versagt, die weiteren Kanten nicht mehr getestet werden müssen. Im ungünstigen Fall müssten pro Dreieck dennoch alle drei Kanten getestet werden. Theoretisch ließe sich dieses Vorgehen auf alle konvexen Polygone verallgemeinern, allerdings wird der Test für Polygone mit vielen Kanten schnell sehr ineffizient.

Strahlmethode nach Jordan: Ein weit verbreiteter "Punkt-in-Polygon"-Test ist die **Strahlmethode nach Jordan** (https://de.wikipedia.org/wiki/Punkt-in-Polygon-Test_nach_Jordan; Jeff Erickson: The Jordan Polygon Theorem. In: Computational Topology. Vorlesungsskript. 2009), in Bonham-Carter (1994,[BC94]) auch allg. als "*point-in-polygon*"-Test bezeichnet. Er basiert auf dem so genannten Jordanschen Kurvensatz (https://de.wikipedia.org/wiki/Jordanscher_Kurvensatz) und lässt sich vergleichsweise effizient auch für nicht-konvexe Polygone mit beliebig vielen Kanten durchführen.

Es soll wieder geprüft werden, ob sich ein Punkt $\vec{x}$ im Inneren eines Polygons $P = (\vec{p_1}, \vec{p_2}, \ldots)$ befindet. Dafür wird gezählt, wie oft ein Strahl mit beliebiger Richtung $\vec{d}$ ausgehend von $\vec{x}$ die Kanten von P schneidet. Ob ein Schnitt mit der Kante $(\vec{p_i}, \vec{p_{i+1}})$ vorliegt, lässt sich aus der folgenden Gleichung ermitteln:

$$\vec{x} + \lambda_1 \cdot \vec{d} = \vec{p_i} + \lambda_2 \cdot (\vec{p_{i+1}} - \vec{p_i}).$$

Ein Schnitt liegt vor, wenn gilt:

$$\lambda_1 > 0 \text{ und } 0 > \lambda_2 \leq 1.$$

Liegen **keine oder eine gerade Anzahl** von Schnitten vor, befindet sich $\vec{x}$ immer **außerhalb** des Polygons. Bei einer **ungeraden Anzahl** von geschnittenen Kanten liegt $\vec{x}$ immer **innerhalb** des Polygons. Dies gilt im Allgemeinen für jedes beliebige Polygon und jede Strahlrichtung. In Ausnahmefällen kann es zu unendlich vielen Schnittpunkten kommen. Dies ist dann der Fall, wenn der Strahl direkt auf einer Polygonkante entlang verläuft. In diesem Fall muss eine andere Richtung $\vec{d}$ gewählt und der Test wiederholt werden.

In der folgenden Abbildung ist dies nochmals verdeutlicht. Der Punkt a_1 liegt im Inneren des grauen, nicht-konvexen Polygons. Der von ihm ausgehende Strahl (gestrichelte schwarze Linie) schneidet die Polygonkanten dreimal (grüne Kreuze), also eine ungerade Anzahl. Punkt a_2 liegt außerhalb des grauen Polygons. Der von ihm ausgehende Strahl schneidet die Polygonkanten viermal (rote Kreuze), also eine gerade Anzahl.

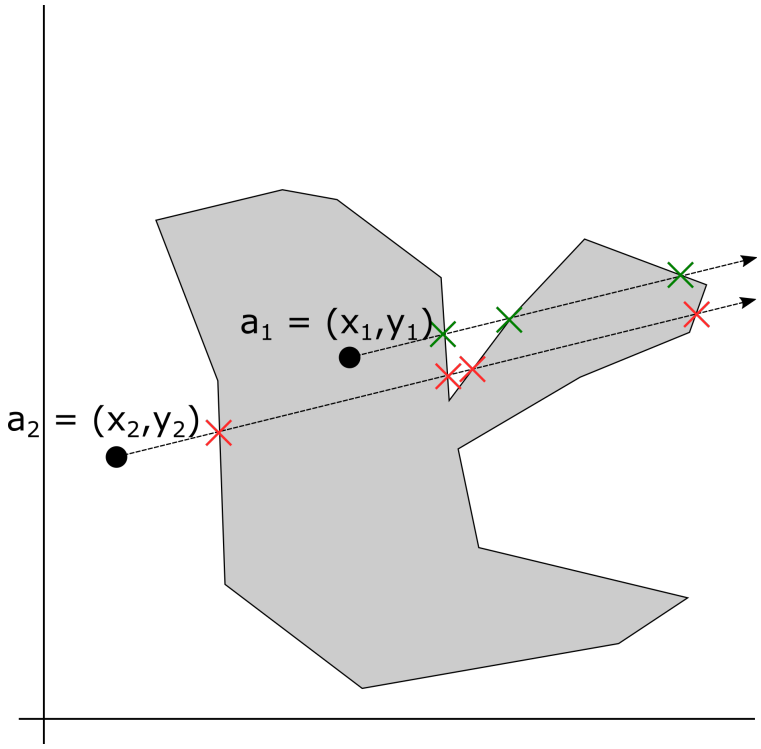


Abbildung 93: Strahlmethode nach Jordan

Auch dieser "Punkt-in-Polygon"-Test lässt sich sehr effizient anwenden. Es müssen immer alle Polygonkanten auf Schnitt getestet werden. Da für jeden Schnitttest aber nur 2 sehr einfache Gleichungen für die beiden Unbekannten λ_1 und λ_2 zu lösen sind, ist jeder einzelne Schnitttest nicht sehr aufwändig zu berechnen.

Fläche-zu-Fläche-Transformationen

Eine weiteres häufiges Problem ist das Übertragen der Attributwerte eines oder mehrerer Flächenobjekte auf ein neues Flächenobjekt. Gegeben sei eine Karte $\mathbf{A}$ bestehend aus n_A Polygonen A_k mit $1 \leq k \leq n_A$. Für jedes Polygon ist ein konstanter Attributwert $f(A_k)$ bekannt. Gegeben sei weiterhin ein weiteres Polygon B. Für B ist kein Wert für ein Attribut bekannt, dieser soll basierend auf den bekannten Werten $f(\mathbf{A})$ bestimmt werden.

In einem ersten Schritt wird die Karte $\mathbf{A}$ mit dem Polygon B verschnitten. Dadurch entsteht eine neue Karte

$$\mathbf{C} = (C_1, \ldots, C_i, \ldots, C_{n_C}) \text{ mit } C_i = A_k \cap B,$$

welche die Schnittpolygone zwischen A_k und B beinhaltet mit $B = \bigcup_{i=1,\ldots,n_c} C_i$. Für jedes Schnittpolygon C_i ist der Attributwert $f(C_i)$ bekannt, dieser entspricht dem Attributwert $f(A_k)$ des ihm zugrunde liegenden Polygons A_k .

Der Attributwert $f(B)$ kann jetzt über ein flächen-gewichtetes Mittel berechnet werden:

$$f(B) = \frac{1}{\text{Fläche}(B)} \sum_{i=1}^{n_c} \text{Fläche}(C_i) \cdot f(C_i).$$

Dies ist schematisch in der folgenden Abbildung dargestellt. Die Farben entsprechen den bekannten Attributwerten.

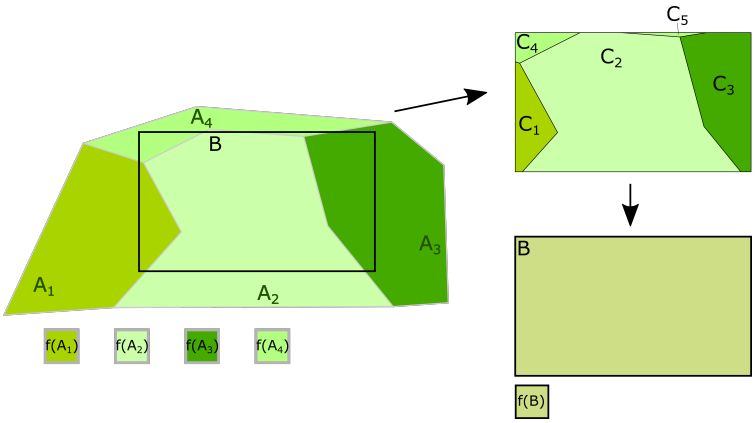


Abbildung 94: Konzept zur Fläche-zu-Fläche-Transformation

Resampling von Geoobjekten

Bei *resampling*-Verfahren wird zwischen *upsampling* und *downsampling* unterschieden. Beim upsampling wird die Auflösung durch Hinzufügen von zusätzlichen Datenpunkten erhöht. Dies kann zum Einen durch die Hinzunahme *zusätzlich gemessener Daten* erfolgen, wenn die vorhandenen Daten für eine Anwendung nicht ausreichen. Ein solches Hinzufügen von neuen Daten **erhöht den Informationsgehalt** eines Datensatzes und kann so die Datenqualität und die Aussagekraft von GIS-Auswertungen erhöhen. Zum Anderen können zusätzlich Punkte interpolativ vorhergesagt und dem initialen Datensatz hinzugefügt werden. Dadurch wird zwar die Auflösung erhöht, es findet aber **kein** Informationszuwachs statt, da die neuen Punkte nur basierend auf den vorhanden Punkten vorhergesagt werden. Ein solches upsampling wird sehr häufig für die "bessere" Visualisierung von Datensätzen verwendet.

Downsampling ist die Gegenoperation zum upsampling und dient dazu, Punkte aus einem Datensatz zu entfernen. In der realen Anwendung kann es vorkommen, dass ein Datensatz mehr Datenpunkte beinhaltet, als durch seinen Informationsgehalt gerechtfertigt ist. Solche Datensätze werden als **oversampled** ("überaufgelöst") bezeichnet. Durch ein downsampling-Verfahren wird versucht, Datenpunkte ohne Beitrag zur Gesamtinformation zu identifizieren und zu entfernen. Häufiger stellt sich allerdings das Problem, dass ein Datensatz so viele Datenpunkte beinhaltet, dass er technisch nur noch schwer zu handhaben ist. Hier ist das Ziel eines downsamplings, die Gesamtpunktanzahl auf ein technisch handhabbares Maß zu reduzieren, ohne zu viele Informationen zu verlieren.

Downsampling kann ebenfalls interpolativ erfolgen, indem an neuen Datenpositionen basierend auf den vorhandenen Daten die Attributwerte vorhergesagt werden. Besser ist jedoch **selektives *downsampling***, bei dem ein Teil der vorhandenen Datenpunkte einfach aus dem Datensatz zu entfernen wird. Dadurch kann wenigstens die Information der nicht entfernten Punkte unverfälscht beibehalten werden. Solche selektiven downsampling-Verfahren lassen sich auch unter dem Begriff *weeding* (engl. für jäten / herausreißen) zusammenfassen. Hier ist es das Ziel, anhand von vorgegebenen Kriterien Punkte aus einer gegebenen Punktmenge zu entfernen. Die Annahme ist, dass die entfernten Punkte gemäß der angesetzten Kriterien zu keiner signifikanten Zusatzinformation für die Gesamtpunktmenge beitragen. Der Vorteil ist, dass man die Datenmenge eines Objektes reduzieren kann, ohne zu viele Informationen zu verlieren. Die Kriterien für das Ausdünnen der Daten können rein geometrisch sein, zum Beispiel die Lage der Punkte, oder sich zusätzlich auf Attributwerte beziehen.

Im Folgenden wird ein Beispiel für einen solchen *weeding*-Algorithmus näher erläutert. Es handelt sich um den so genannten **Douglas-Peucker-Algorithmus** (https://de.wikipedia.org/wiki/Douglas-Peucker-Algorithmus, [BC94]) zur Vereinfachung (Generalisierung) hochaufgelöster Linienobjekte. Dieser Algorithmus betrachtet nur die Lage der Punkte und berücksichtigt keine eventuell vorhandenen Attribute.

Ausgangspunkt ist ein hoch aufgelöstes Linienobjekt als Sequenz von m Punkten $(p_1, \ldots, p_m)$, zusätzlich wird ein Toleranzwert $\epsilon > 0$ vorgegeben.

In einem ersten Schritt werden der erste und der letzte Punkt des Linienobjekte durch eine

Linie $L_{1,m}$ verbunden. Dann wird der senkrechte Abstand d aller anderen Punkte $(p_2, \ldots, p_{m-1})$ zu dieser Linie ermittelt. Der Punkt p_i mit dem maximalen Abstand zu $L_{1,m}$, für den also gilt

$$d(p_i, L_{1,m}) = \max(d(p_j, L_{1,m})), j = 2, \ldots, i, \ldots, m-1$$

unterteilt die Linie $L_{1,m}$ in zwei Liniensegmente $L_{1,i}$ und $L_{i,m}$. Für jedes dieser Liniensegmente wird diese sukzessive Unterteilung fortgesetzt, solange Punkte zwischen Anfangs- und Endpunkt des Liniensegments vorliegen, welche einen Abstand $d > \epsilon$ zu einem Segment aufweisen. Alle Punkte, welche durch dieses Verfahren aus der initialen Punktmenge entfernt werden, weisen einen Abstand zum neuen Linienobjekt (bestehend aus den neu gebildeten Liniensegmenten) auf, der **immer** kleiner als der Toleranzwert ϵ ist.
Das konzeptuelle Vorgehen für *weeding* mittels des Douglas-Peucker-Algorithmus wird in der folgenden Abbildung nochmals schematisch erklärt.

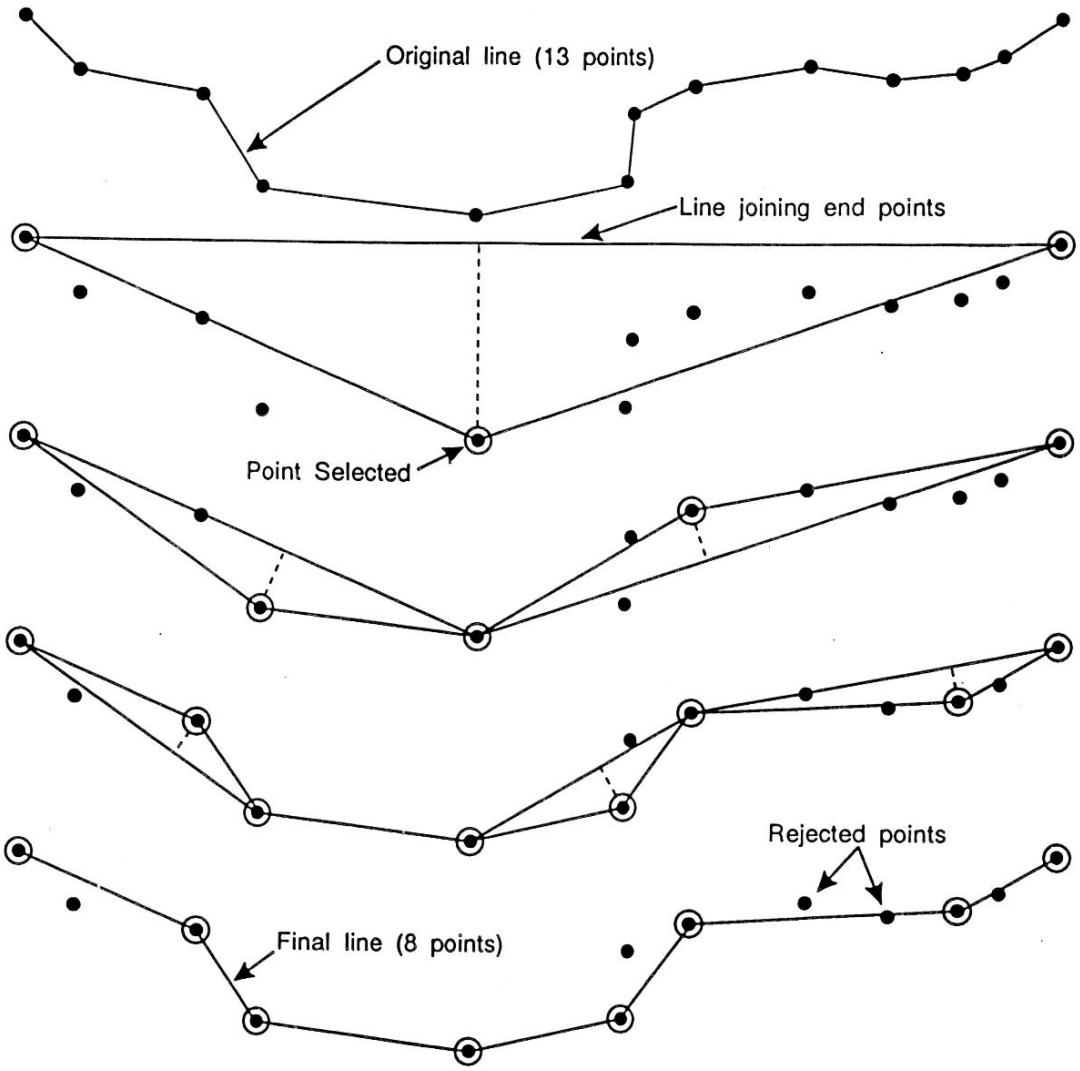


FIG. 6-10. Douglas-Peucker algorithm for line generalization. The original line contains 13 points, and the final line contains 8 points. The rejected points are said to be "weeded". A number of alternative algorithms for line weeding have been proposed.

Abbildung 95: Konzept für *weeding* basierend auf Douglas-Peucker-Algorithmus ([BC94])

7.3 Transformationen zur Änderung von Form und Ausdehnung

Buffering

Beim ***buffering*** (alternativ auch *dilation*, *spreading*) wird die räumliche Ausdehnung von Objekten virtuell erweitert. Dabei werden so genannte **Buffer-Zonen** $B(O)$ um ein räumliches Objekt O erzeugt. Es handelt sich um Regionen relativer Nähe (*proximity zones*) zu diesem Objekt. Ein beliebiger Punkt $\vec{x}$ befindet sich innerhalb einer Zone $B_r(O)$ um ein Objekt O, wenn gilt:

$$\vec{x} \in B_r(O) \rightarrow d(\vec{x}, O) \leq r.$$

r ist dabei die max. Ausdehnung der Buffer-Zone und $d(\vec{x}, O)$ der Abstand des Punktes vom Objekt. Ein so genannter **Buffer-Korridor** $B_{r_1,r_2}(O)$ umfasst alle Punkt $\vec{x}$ um ein Objekt, für die gilt: $\vec{x} \in Br_1, r_2(O) \rightarrow r_1 \leq d(\vec{x}, O) \leq r_2$, die also in einem Bereich um das Objekt liegen, welche mindestens einen Abstand r_1 und maximal einen Abstand r_2 zu diesem Objekt aufweist.

Diese Buffer-Zonen / -Korridore liegen zumeist nur "virtuell" vor. Es handelt sich zwar immer im flächenhaft ausgedehnte Regionen, es handelt sich aber nicht um eigenständige Geoobjekte. Zur Überprüfung, ob sich ein Objekte O' innerhalb einer solchen Zonen $Br_1, r_2(O)$ befindet, ist es aber nur notwendig, den Abstand $d(O', O)$ zwischen beiden Objekten zu bestimmen. Die Analyse solcher "Nähe"-Beziehungen ist also sehr effizient möglich.

Beispiele für Buffer-Zohnen/-Korridore um Punkt-, Linien- und Flächenobjekte sind in der folgenden Abbildung schematisch dargestellt.

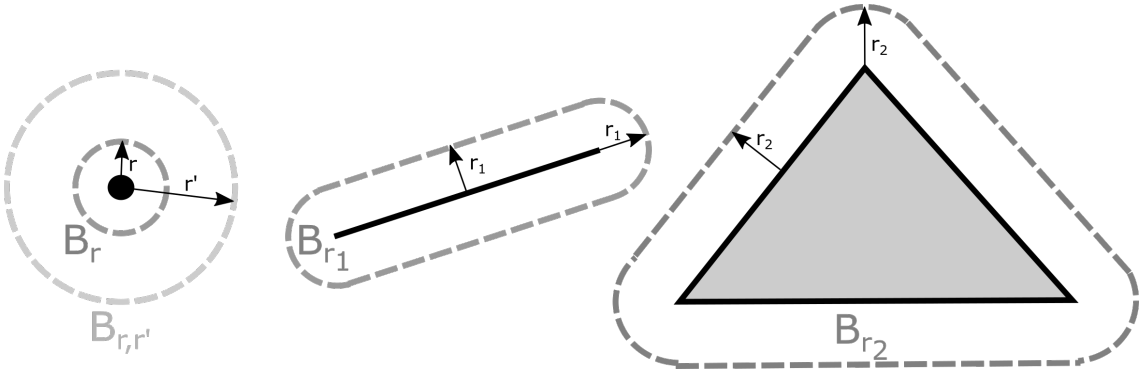


Abbildung 96: Verschiedene Buffer-Zohnen (grau, gestrichelt) um Punkt-, Linien und Flächenobjekte

In der nachfolgenden Abbildung werden Buffer-Korridore verwendet, um die "Nähe"-Beziehungen zwischen Goldvorkommen (Punkte) in einer Region zu den Achsen von verfalteten Gesteinsschichten (Linien) zu illustrieren. Goldvorkommen zeichnen sich dabei häufig durch relative "Nähe" zu den Falten-Achsen aus.

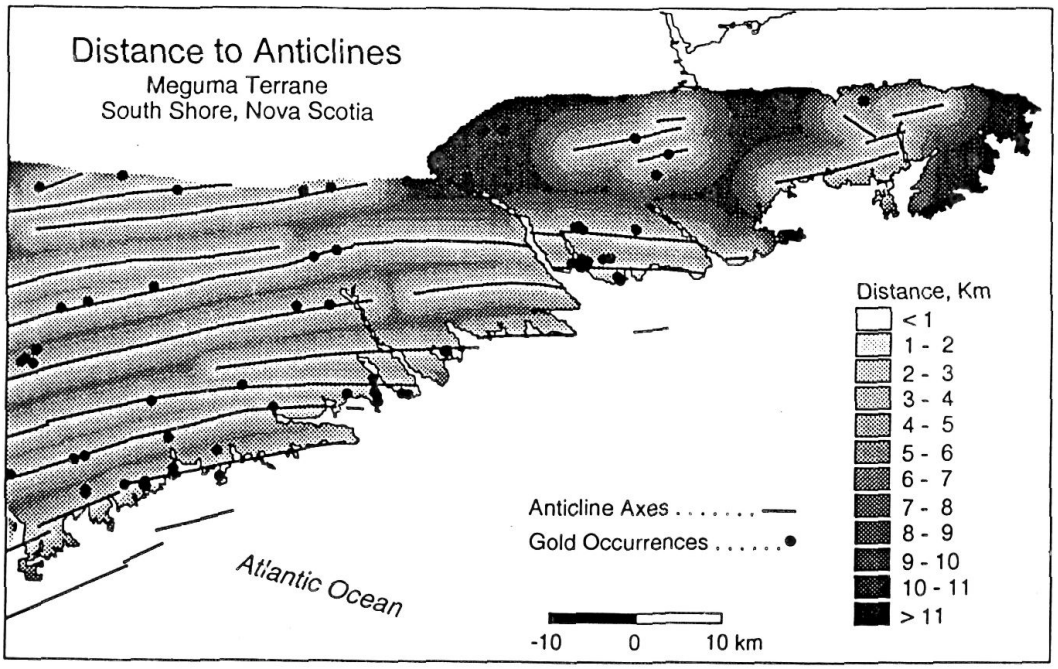


FIG. 6-8. A. Map to illustrate the dilation or buffering of linear features. Anticlinal fold axes in Meguma terrane, Nova Scotia have been successively dilated with corridors (250 m intervals) to produce a map showing proximity to the nearest fold axis. A classification has been applied so that the intervals on the map are 1 km. The points are locations of gold occurrences.

Abbildung 97: Anwendungsbeispiel für Buffer-Zonen aus Bonham-Carter (1994, [BC94])

Räumliche Filterung

Bei **räumlicher Filterung** handelt es sich um Operationen, um in **kontinuierlichen Rasterdaten** räumliche Strukturen zu verstärken oder zu unterdrücken. Es handelt sich dabei immer um **lokale** Operationen, welche den Wert an einem Pixel basierend auf den Werten benachbarter Pixel verändern.

Mathematisch lässt sich eine solche Filteroperation als **Faltung** (Konvolution, *convolution*) zweier Funktionen f und k beschreiben. Diese ist wie folgt definiert:

$$(f * k)(\vec{x}) := \int f(\vec{u})k(\vec{x} - \vec{u})d\vec{u} \approx \sum_{l=1^p} f(\vec{u_l})k(\vec{x} - \vec{u_l})\Delta u.$$

Das Ergebnis einer Faltung besitzt dabei immer die "besseren" Eigenschaften der beiden beteiligten Funktionen, d. h. wenn eine der Funktionen nur stetig ist, die andere aber stetig differenzierbar, dann ist das Faltungsergebnis auch stetig differenzierbar. Des Weiteren gilt:

$$\mathcal{F}(f * k) = \mathcal{F}f \cdot \mathcal{F}k.$$

Eine Funktion $k_h(\vec{x})$ mit kompaktem Support D_h weist nur innerhalb eines begrenzten Gebietes D_h Werte ungleich Null auf mit

$$k_h(\vec{x}) \equiv 0 \quad \forall \vec{x} \notin D_h.$$

Ist diese Funktion zusätzlich "glatt", wird sie als so genannte **Kernelfunktion** bezeichnet und für eine Faltung mit dieser Funktion gilt

$$(f * k_h)(\vec{x}) = \int f(\vec{u})k_h(\vec{x} - \vec{u})d\vec{u} \to f(\vec{x}), h \to 0,$$

das heißt, die Faltung bildet die Funktion f auf sich selbst ab, wenn der Support der Kernelfunktion sehr klein wird bzw. verschwindet. Die Faltung einer reellwertigen Funktion $f \to R$ mit einer radialsymmetrischen Kernelfunktion k_h mit kompaktem Support D_h erfüllt zudem folgende Bedingung:

$$\begin{aligned}(f * k_h)(\vec{x}) &= \int_R f(\vec{u})k_h(\vec{x}-\vec{u})d\vec{u} \\ &= \int_{\vec{x}-\vec{u}\in D_h} f(\vec{u})k_h(\vec{x}-\vec{u})d\vec{u} \\ &= \int_{\vec{u}\in D_h} f(\vec{u})k_h(\vec{x}-\vec{u})d\vec{u} \\ &= \int_{\vec{u}\in D_h} f(\vec{x}-\vec{u})k_h(\vec{u})d\vec{u}\end{aligned}$$

.

Für $D_h = [-h, h]$ kann man dann schreiben

$$(f * k_h)(x) \approx \sum_{l=1}^{p} f(x-u_l)k(u_l)\Delta u = \sum_{u_l\in[-h,h]} f(x-u_l)k(u_l)\Delta u \text{ mit } x-u_l \in [x-h, x+h].$$

Räumliche Filteroperationen sind definiert über die pixelweise diskrete Faltung der Funktion der Pixelwerte eines Rasterbildes mit einem Strukturelement. Dieses ist die diskrete Version eine radialsymmetrischen Kernelfunktion. Für die Pixel eines diskreten Rasterbildes wird eine Faltung wie folgt repräsentiert

$$\begin{aligned}(f * k_h)(p(i_0, j_0)) &= \sum_{p(i,j)\in d_h(p(i_0,j_0)),\ p'(i,j)\in d_h} f(p(i,j))k_h(p'(i,j)) \\ &= \sum_{i=-m}^{m}\sum_{i=-n}^{n} f(p(i_0+i, j_0+j))k_{2m+1,\ 2n+1}(p(i,j))\end{aligned}$$

.

Häufig gilt $m = n$ mit kleinen Werten für $m = 1,\ 2,\ 3$. Umfasst d_h zum Beispiel nur die echte Nachbarschaft (siehe Abschnitt Rastermodell) eines Pixels $p(i_0, j_0)$, vereinfacht sich der allgemeine Fall zu

$$\begin{aligned}(f * k_h)(p(i_0, j_0)) =& f(p(i_0, j_0))k_h(0, 0) \\ &+ f(p(i_0, j_0-1))k_h(0, -1) + f(p(i_0, j_0+1))k_h(0, 1) \\ &+ f(p(i_0-1, j_0))k_h(-1, 0) + f(p(i_0+1, j_0))k_h(1, 0)\end{aligned}$$

.

Ein diskrete Faltung über einem Rasterbild kann auch als gewichtetes Mittel über die benachbarten Pixelwerte eines Pixels angesehen werden. Die Gewichte ergeben sich aus den Wertes des Strukturelements / der Kernelfunktion.

Der Begriff ***räumliche Frequenz*** (*spatial frequency*) beschreibt die lokale Variation der Pixelwerte in einer (kleinen) Nachbarschaft um einem Pixel. Ist diese "niedrig" (low), sind die lokalen Änderungen der Pixelwerte zwischen benachbarten Pixel gering. Die Pixelwerte innerhalb einer Nachbarschaft sind vergleichsweise gleichmäßig. Ist die räumliche Frequenz dagegen "hoch" (high), variieren die Werte zwischen benachbarten Pixel stark. Die Pixelwerte innerhalb einer Nachbarschaft sind also eher heterogen.

***Low-pass* Filter:** Räumliche *low-pass* Filter $\mathcal{L}$ verstärken niedrige räumliche Frequenzen und unterdrücken lokal extreme Werte / starke Variabilitäten. Die Funktion aller Pixelwerte wird dadurch geglättet (Glättungsfilter). Beispiele für von Glättungsfiltern verwendete Strukturelemente sind

$$k^1_{3,3} = \frac{1}{5}\begin{pmatrix}0&1&0\\1&1&1\\0&1&0\end{pmatrix},\ k^2_{3,3} = \frac{1}{9}\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix} \text{ oder } k^3_{3,3} = \frac{1}{15}\begin{pmatrix}1&2&1\\2&3&2\\1&2&1\end{pmatrix}.$$

Es wird immer der Pixelwerte $f(p(i_0, j_0))$ durch das gewichtete Mittel seiner Nachbarn ersetzt. Dabei glättet $k^3_{3,3}$ weit weniger stark als die beiden anderen Strukturelemente. Die nachfolgende Animation verdeutlicht die Wirkungsweise eines solchen Filters, als Filterkernel wurde $k^1_{3,3}$ verwendet. Lokal stark variierende Pixelwerte werden dadurch geglättet.

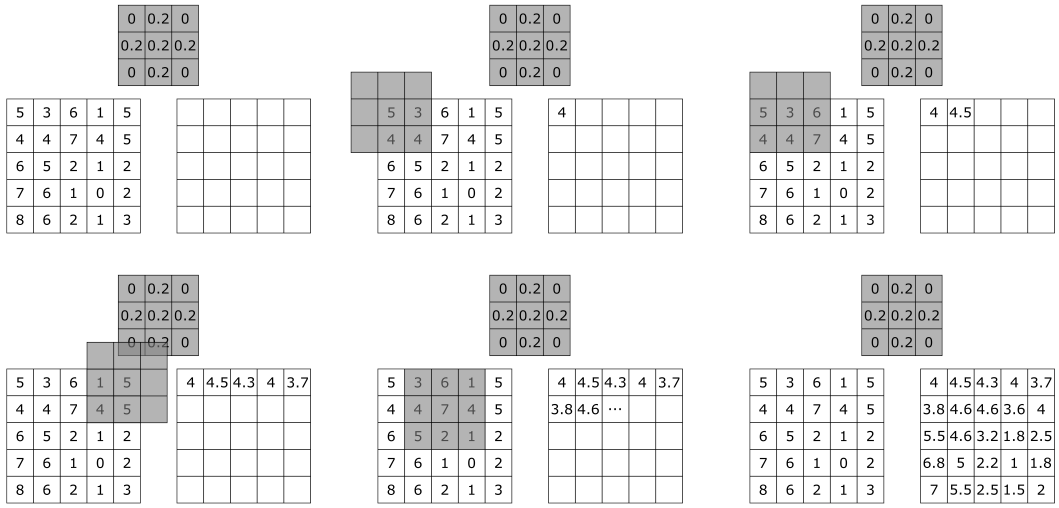


Abbildung 98: Wirkungsweise eines Glättungsfilters an einem einfachen Beispielraster

Filter zur Korrektur fehlender Werte: Das Strukturelement $k_{3,3} = \frac{1}{4}\begin{pmatrix}0&1&0\\1&0&1\\0&1&0\end{pmatrix}$ kann zur Korrektur fehlender oder fehlerhafter Werte verwendet werden. Der fehlende Pixelwert wird durch den Mittelwert seiner echten Nachbarn ersetzt.

***High-pass* Filter / Filter zur Kantenverstärkung:** Räumliche *high-pass* Filter $\mathcal{H}$ verstärken die lokale Variabilität in einem Rasterbild. Sie werden im Allgemeinen konstruiert, indem von einem Bild eine über einen low-pass Filter erzeugte Version des Bildes abgezogen wird mit

$$(\mathcal{H}f)(p(i_0, j_0)) = f(p(i_0, j_0)) - (\mathcal{L}f)(p(i_0, j_0)).$$

Durch einen solchen high-pass Filter lassen sich zum Beispiel Kanten in einem Rasterbild verstärken (*edge enhancement*), indem zum initialen Bild eine über einen high-pass Filter erzeugte Version des Bildes addiert wird mit

$$f(p(i_0, j_0)) + (\mathcal{H}f)(p(i_0, j_0)) = 2f(p(i_0, j_0)) - (\mathcal{L}f)(p(i_0, j_0)).$$

Andere gängige Filter zur Verstärkung von Kanten sind der **Laplace-Filter** $k^{\text{Laplace}}_{3,3} = \begin{pmatrix}0&1&0\\1&-4&1\\0&1&0\end{pmatrix}$ oder die so genannten **Sobel-Filter** $k^{\text{Sobel x}}_{3,3} = \begin{pmatrix}-1&0&1\\-2&0&2\\-1&0&1\end{pmatrix}$ und $k^{\text{Sobel y}}_{3,3} = \begin{pmatrix}-1&-2&-1\\0&0&0\\1&2&1\end{pmatrix}$.

Die Sobel-Filer erzeugen zwei Ausgabebilder, bei denen das Erste, basierend auf $k_{3,3}^{\text{Sobel x}}$ die erste räumliche Ableitung in x-Richtung $\frac{\partial f}{\partial x}$ und das Zweite, basierend auf $k_{3,3}^{\text{Sobel y}}$, die erste räumliche Ableitung in y-Richtung $\frac{\partial f}{\partial y}$ des Bildes repräsentiert. Durch die Kombination beider Ergebnisse lässt sich zum Beispiel die Magnitude des lokalen Gradienten (*slope*) bestimmen mit

$$\text{slope} = \| \bigtriangledown f \| = \sqrt{\frac{\partial f^2}{\partial x} + \frac{\partial f^2}{y}}.$$

Räumliche Texturfilter: Texturfilter erlauben es, ein Maß für die "Textur" eines Bildes anzugeben. Dies umfasst unter anderem die lokale "Rauheit" der Pixelwerte.

$$\text{Lokale Varianz: Varianz}(p(i_0, j_0)) = \sum_{i=-m}^{m} \sum_{i=-n}^{n} \frac{\left(f(p(i_0, j_0)) - \bar{f}(p(i_0, j_0))\right)^2}{mn}$$

mit $\bar{f}(p(i_0, j_0))$ als Mittelwert aller Pixelwerte in der Nachbarschaft von $p(i_0, j_0)$.

$$\text{Lokale Entropie: } S = \sum_{i=-m}^{m} \sum_{i=-n}^{n} (f(p(i_0, j_0)) \ln f(p(i_0, j_0))).$$

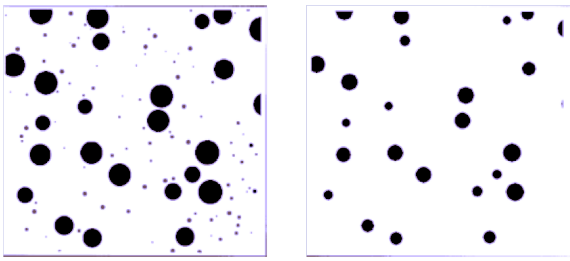
Nicht-konvolutionelle Filter: Eine Reihe von Filtern lassen sich nicht mathematisch als Faltung ausdrücken. Ein sehr einfaches Beispiel hierfür ist der so genannte **Median-Filter**, welcher einen Pixelwert durch den Median aller Pixelwerte in der Nachbarschaft ersetzt. Eine Faltung kann Operationen wie Addition und Multiplikation repräsentieren, weswegen ein **Mittelwert-Filter** über ein Faltung dargestellt werden kann. Zur Berechnung des Medians benötigt allerdings die Sortierung und das Abzählen der Pixelwerte in der Nachbarschaft. Dies lässt sich in keinem Fall über eine Faltung ausdrücken.

Morphologische Operatoren

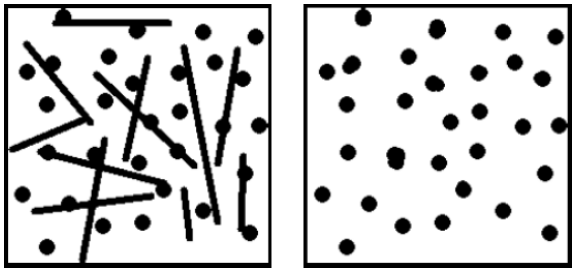
In einem binären Rasterbild nehmen die Pixelwerte nur zwei unterschiedliche Werte an, zumeist 0 / weiß und 1 / schwarz. Räumliche Objekte in einem solchen Rasterbild sind zusammenhängende Gruppen von gleichartiger Pixel. In der Praxis wird ein solches Rasterbild über eine rechteckige Matrix A repräsentiert, deren Einträge entweder 0 oder 1 sind. Im Folgenden sind die Objekte **immer über schwarze Pixel** dargestellt. Im Allgemeinen hängt dies von der gerade verwendeten Definition ab, d.h. je nach Anwendung können Objekte auch über weiße Pixel und die Umgebung über schwarze Pixel repräsentiert werden.

Bei Objekten in binären Bildern treten gegebenen Falls Situationen auf, welche durch darauf angepasste Operationen bearbeitet werden müssen:

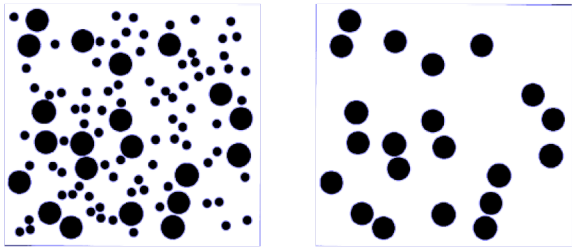
- Rauschen: sehr viele kleine Objekte → zu kleine Objekte müssen entfernt werden;



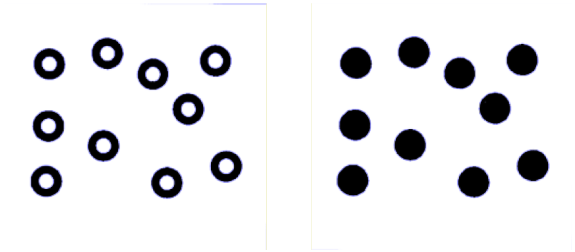
- Entfernen unerwünschter Objekte, z.b. bzgl. Größe oder Form;



- Klassifikation nach Objektgrößen;



- Löcher in Objekten → Schließen der Löcher;



- oder zum Beispiel lokale Gruppen von nahen kleinen getrennten Objekten → zusammenfassen dieser Gruppen zu einen größeren Objekt.

Mathematische Morphologie: Die oben beschriebenen Operationen lassen sich mittels **mathematischer Morphologie** durchführen. Ein Geoobjekt X besteht aus einer Menge benachbarter schwarzer Pixel p. Eine morphologische Operation wird auf dieser (morphologischen) Menge X durchgeführt.

Die einfachste morphologische Operation ist die so genannte **Translation**, also die Verschiebung der Menge $X \in A$ um einen Vektor $\vec{t}$. Diese ist folgender maßen definiert:

$$X_t = \{p | p - \vec{t} \in X\}.$$

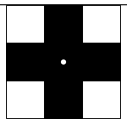
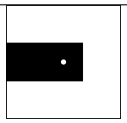
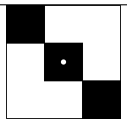
Das verschobene Objekte / die verschobene Menge X_t besteht aus allen Pixeln $p + \vec{t}$, wenn p ein Pixel der Menge ist mit $p \in X$. Durch diese Translation wird ein neues Bild C erzeugt, welches die gleiche Größe hat wie die Bildmatrix A, das verschobene Objekt ist ein Teil dieser Bildmatrix mit $X_t \in C$. Einem Pixelwert $f(p + \vec{t})$ am Pixel $p + \vec{t}$ im Bild C wird der Pixelwert $f(p)$ für den Pixel p aus der Bildmatrix A zugewiesen.

Die Operation **Punkt-Inversion** ist definiert durch $\check{X} := \{-p | p \in X\}$ erzeugt eine an einer Achse gespiegelte Version $\check{X}$ des Objektes X. Ist das Objekt X symmetrisch, dann gilt $\check{X} = X$.

Neben dem Objekt $X \in A$ wird für weitergehende Operationen ein so genannter **(morphologischer) Operator** B benötigt. Dieser wird auch als Strukturelement bezeichnet. Es handelt sich hierbei ebenfalls um eine morphologische Menge, d.h. die oben beschrieben Operationen Translation und Punkt-Inversion gelten analog auch für B. Es handelt sich Strukturelementen zumeist

um quadratische Matrizen oder Pixelmasken mit einer ungeraden Anzahl von Zeilen und Spalten, welche (ähnlich zu dem Kernelmatrizen bei der Bildfilterung) über jedem Matrixeintrag / Pixel des Bildes A platziert werden können. Im Gegensatz zur Bildmatrix beinhalten sie entweder den Wert 1 (schwarz) oder den Wert NULL, also keinen Wert. Im Folgen sind mögliche Beispiele für Strukturelemente dargestellt. Das zentrale Element (Referenzpixel) ist entweder über eine fett dargestellte Zahl oder einen kleinen weißen Kreis dargestellt:

Tabelle 20: Beispiele für Strukturelemente

$B_1 = \begin{pmatrix} \text{NULL} & 1 & \text{NULL} \\ 1 & \mathbf{1} & 1 \\ \text{NULL} & 1 & \text{NULL} \end{pmatrix}$	
$B_2 = \begin{pmatrix} \text{NULL} & \text{NULL} & \text{NULL} \\ 1 & \mathbf{1} & \text{NULL} \\ \text{NULL} & \text{NULL} & \text{NULL} \end{pmatrix}$	
$B_3 = \begin{pmatrix} 1 & \text{NULL} & \text{NULL} \\ \text{NULL} & \mathbf{1} & \text{NULL} \\ \text{NULL} & \text{NULL} & 1 \end{pmatrix}$	

Morphologische Erosion: Die **morphologische Erosion** $\epsilon_B(X)$ (*erosion*) eines Objektes X bezüglich eines Strukturelements B ist definiert als die Menge aller Pixel $p \in X$ für die das über diesem Pixel zentrierte Strukturelement B_p komplett im Objekt X enthalten ist. Das heißt, wenn das Referenzpixel des Strukturelements auf dem Pixel p liegt, müssen alle Pixel aus B, für die ein Wert existiert ($\neq$ NULL), über einem Pixel liegen, welches zum Objekt X gehört:

$$\epsilon_B(X) := \{p | B_p \subset X\}.$$

Sei $\vec{b} \in B$ ein Differenzvektor zwischen einem Pixel von B, dessen Wert existiert und dem Referenzpixel. Für das oben gezeigte Strukturelement B_1 würde dies wie folgt aussehen:

$$B_1 = \begin{pmatrix} \text{NULL} & 1 & \text{NULL} \\ 1 & \mathbf{1} & 1 \\ \text{NULL} & 1 & \text{NULL} \end{pmatrix} \rightarrow \begin{pmatrix} \text{NULL} & \vec{b}_{01-} = (0,-1) & \text{NULL} \\ \vec{b}_{-10} = (-1,0) & \vec{b}_{00} = (0,0) & \vec{b}_{10} = (1,0) \\ \text{NULL} & \vec{b}_{01} = (0,1) & \text{NULL} \end{pmatrix}.$$

Die b-Translation X_{-b} ist die Verschiebung des Objekts X um ein $\vec{b} \in B$. Eine Erosion lässt sich diesbezüglich auch als die Schnittmenge aller b-Translationen von X ausdrücken:

$$\epsilon_B(X) = \bigcap_{\vec{b} \in B} X_{-b}.$$

Wenn das Strukturelement **symmetrisch** ist mit $\check{B} = B$, lässt sich die Erosion einer Bildmatrix $\delta_B(A)$ auch als **Minkowski-Differenz** zwischen der Bildmatrix A und dem Strukturelement B schreiben mit $C = \epsilon_B(A) = A \ominus B$.

In der Praxis beinhaltet $\epsilon_B(X) \in C$ weniger Pixel als $X \in A$. Die Erosion entfernt Pixel, welche sich am Rand von X befinden, das Objekt wird dadurch kleiner. Dies ist exemplarisch in der folgenden Animation dargestellt. Das Strukturelement (oben) wandert sukzessive über das Objekt. Nur wenn alle Teile des Strukturelements innerhalb des Objektes liegen (grüner Kreis), wird in der neuen Bildmatrix ein Pixel mit dem Wert 1 / schwarz besetzt. Ansonsten (rotes X) wird der Pixelwert 0 / weiß belassen.

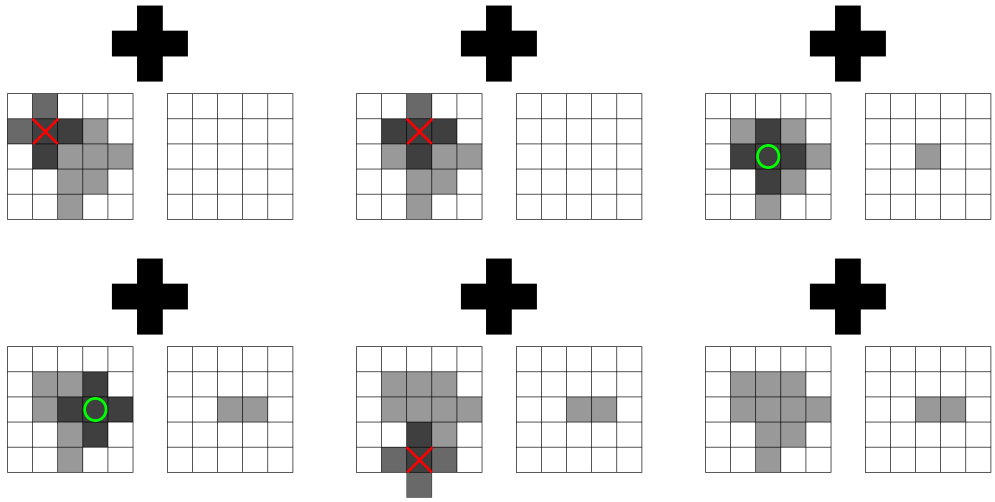


Abbildung 99: Beispiel für Erosion eines Objektes (links) anhand eines Strukturelements (oben). Das Ergebnis ist rechts dargestellt

Morphologische Dilatation: Die **morphologische Dilatation** $\delta_B(X)$ (*dilation*) eines Objektes X bezüglich eines Strukturelements B ist definiert als die Menge aller Pixel $p \in X$ für die mindestens ein Pixel des über diesem Pixel zentrierte Strukturelements B_p im Objekt X enthalten ist. Das heißt, wenn das Referenzpixel des Strukturelements auf dem Pixel p liegt, muss mindestens ein Pixel aus B über einem Pixel liegen, welches zum Objekt X gehört: $\delta_B(X) := \{p | B_p \cap X \neq \emptyset\}$. Alternativ lässt sich eine Dilatation auch als die Vereinigungsmenge aller b-Translationen von X ausdrücken: $\delta_B(X) = \bigcup_{\vec{b} \in B} X_{-b}$. Wenn das Strukturelement **symmetrisch** mit $\check{B} = B$ ist, lässt sich die Dilatation einer Bildmatrix $\delta_B(A)$ auch als **Minkowski-Summe** zwischen der Bildmatrix A und dem Strukturelement B schreiben mit $C = \delta_B(A) = A \oplus B$.

In der Praxis beinhaltet $\delta_B(X) \in C$ mehr Pixel als $X \in A$. Die Dilatation füg am Rand von X neue Pixel zu X hinzu. Das Objekt wird dadurch größer. Dies ist exemplarisch in der folgenden Animation dargestellt. Das Strukturelement (oben) wandert sukzessive über das Bild. Wenn ein Teil des Strukturelements innerhalb des Objektes liegt (grüner Kreis), wird in der neuen Bildmatrix ein Pixel mit dem Wert 1 / schwarz besetzt. Ansonsten (rotes X) wird der Pixelwert 0 / weiß belassen.

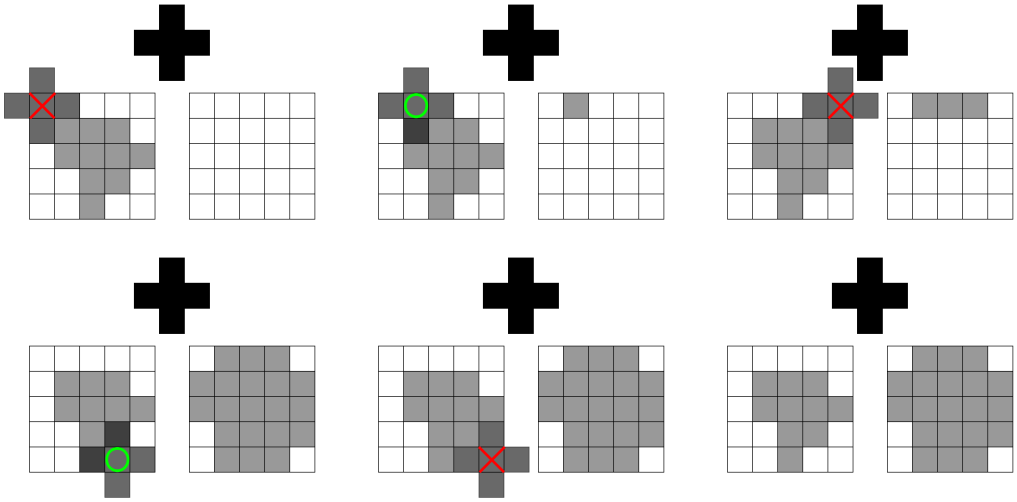


Abbildung 100: Beispiel für Dilatation eines Objektes (links) anhand eines Strukturelements (oben). Das Ergebnis ist rechts dargestellt

Morphologische Operationen für Grauwert-Bilder: Die morphologischen Operationen lassen sich auch für Grauwert-Bilder verallgemeinern. Im Gegensatz zu binären Bildern (nur zwei Werte: 0 / weiß, 1 / schwarz), beinhaltet eine Grauwert-Bildmatrix A alle möglichen Werte zwischen 0 (schwarz) und 1 (weiß) mit $f(A) \rightarrow [0 \geq f \leq 1]$. In diesem Fall muss eine morphologische Operation nicht mehr nur auf dem Objekt, sondern auf dem gesamten Bild durchgeführt werden.

Im Fall der Erosion wird der Pixelwert über das Minimum aller durch das Strukturelement überdeckten Pixelwerte ermittelt:

$$\epsilon_B(A) \rightarrow [\epsilon_B(f)](p) = \min_{\vec{b} \in B} f(p + \vec{b}).$$

Im Fall der Dilation wird anstelle des Minimums das Maximum der überdeckten Pixelwerte verwendet:

$$\delta_B(A) \rightarrow [\delta_B(f)](p) = \max_{\vec{b} \in B} f(p + \vec{b}).$$

Kombinierte Operationen - Gradienten / Abschluss / Öffnung: Grundsätzlich lassen sich morphologische Operationen mehrfach auf das gleiche Objekt / Bild anwenden, auch mehrfache Kombinationen verschiedener Operationen sind möglich.

Im Allgemeinen kann man erwarten, dass die Grenzen von Geoobjekten in einem Bild sich dort befinden, wo sich die Pixelwerte stark ändern. Gradienten-Operatoren dienen dazu, diese Grenzen zu verstärken. **Morphologische Gradienten** verstärken diese Variationen innerhalb des durch das Strukturelement vorgegebenen Fensters. Durch sequenzielle kombinierte Anwendung von Erosion und Dilatation lassen sich diese Gradienten verstärken.

- Arithmetische Differenz zwischen Dilatation und Erosion: $C = \rho_B(A) = \delta_B(A) - \epsilon_B(A)$
- Arithmetische Differenz zwischen Dilatation und Bildmatrix: $C = \rho_B^-(A) = \delta_B(A) - A$
- Arithmetische Differenz zwischen Bildmatrix und Erosion: $C = \rho_B^+(A) = A - \epsilon_B(A)$

Über $\rho_B(A)$ lässt sich näherungsweise die maximale Variation der Grauwerte innerhalb einer lokalen Nachbarschaft ermitteln. $\rho_B^-(A)$ verstärkt die inneren Grenzen von "hellen" (hohe Grauwerte) Objekten, $\rho_B^+(A)$ verstärkt bevorzugt die Außengrenzen von weißen Objekten mit Grauwerten nahe 1.

Bei einer **morphologischen Öffnung** (*opening*) wird eine Dilatation auf das Ergebnis einer Erosion angewendet mit

$$\gamma_B(A) := \bigcup\{B | B \subset X\} := \delta_{\check{B}}[\epsilon_B(A)].$$

Es handelt sich also um die Vereinigungsmenge aller Strukturelemente, welche komplett im ursprünglichen Objekt X enthalten sind. Das Objekt wird zwar verkleinert, aber weniger stark als durch die Erosion allein. Die Öffnung erfüllt zudem folgende Eigenschaften:

- $\gamma_B(X) \subset X \rightarrow \gamma_B(X)$ ist Teilmenge von X
- $\gamma_B[\gamma_B] = \gamma_B \rightarrow$ selbsterhaltend

Die Öffnungsoperation für das oben bereits gezeigte Beispiel ist in de folgenden Abbildung dargestellt.

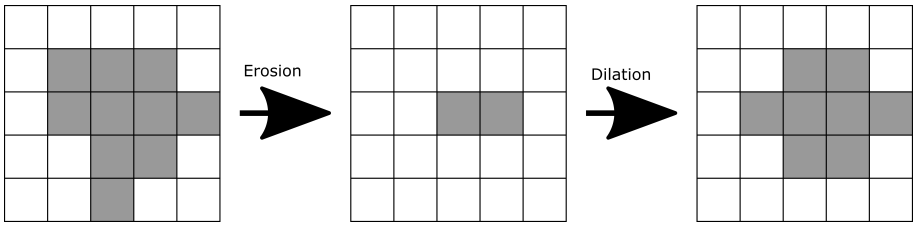


Abbildung 101: Öffnung / Opening für Beispiel

Die morphologische Öffnung wird in der Praxis sehr häufig verwendet, da sie es erlaubt, Rauschen aus einem Binärbild zu entfernen. Dies beruht darauf, dass ein Objekt, welches durch die Erosion komplett ausgelöscht wurde, durch die nachfolgende Dilatation nicht mehr vergrößert / wiederhergestellt werden kann. Die Öffnung kann zudem dazu führen, dass kleine Objekte getrennt werden.

Eine weitere kombinierte Operation ist der **morphologische Abschluss** (*closure*). Hier wird eine Erosion auf das Ergebnis einer Dilation angewendet mit $\phi_B(A) := \epsilon_{\check{B}}[\delta(A)]$. Das Objekt wird zwar vergrößert, aber weniger stark als durch die Dilatation allein. Ein Abschluss erfüllt zudem folgende Eigenschaften:

- $X \subset \phi_B(X) \rightarrow X$ ist Teilmenge von $\phi_B(X)$
- $\phi_B[\phi_B] = \phi_B \rightarrow$ selbsterhaltend

Die Abschlussoperation für das oben bereits gezeigte Beispiel ist in der folgenden Abbildung dargestellt.

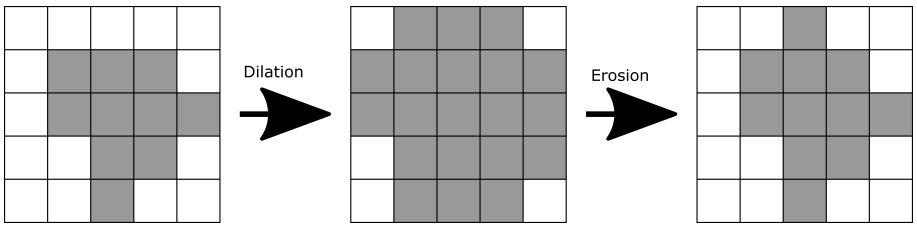


Abbildung 102: Abschluss / Closue für Beispiel

Die Abschlussoperation kann "Löcher" innerhalb von Objekten schießen. Zudem kann der Leerraum zwischen nahen, aber getrennten Objekten aufgefüllt werden. Diese Objekte werden dann zu einem Einzelobjekt verbunden.

Literatur

[Bar05] Norbert Bartelme. *Geoinformatik: Modelle, Strukturen, Funktionen.* Springer, Berlin, 4 edition, 2005.

[BC94] G.F. Bonham-Carter. *Geographic Information Systems for Geoscientists: Modelling with GIS.* Computer methods in the geosciences. Pergamon, 1994.

[Bil10] R. Bill. *Grundlagen der Geoinformationssysteme.* Wichmann, 2010.

[Cre15] Noel A Cressie. *Statistics for Spatial Data, Revised Edition.* Wiley, 2015.

[GAMR15] H.J. Götze, J. Arndt, D. Mertmann, and U. Riller. *Einführung in die Geowissenschaften.* UTB M. UTB GmbH, 2015.

[Mal02] J.L. Mallet. *Geomodeling.* Applied Geostatistics. Oxford University Press, 2002.