TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG

Die Ressourcenuniversität. Seit 1765.

Grundlagen GIS

Sommersemester 2024

Prof. Christian Gerhards
Arbeitsgruppe Geomathematik und Geoinformatik

OPAL:
https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/197584157373?6

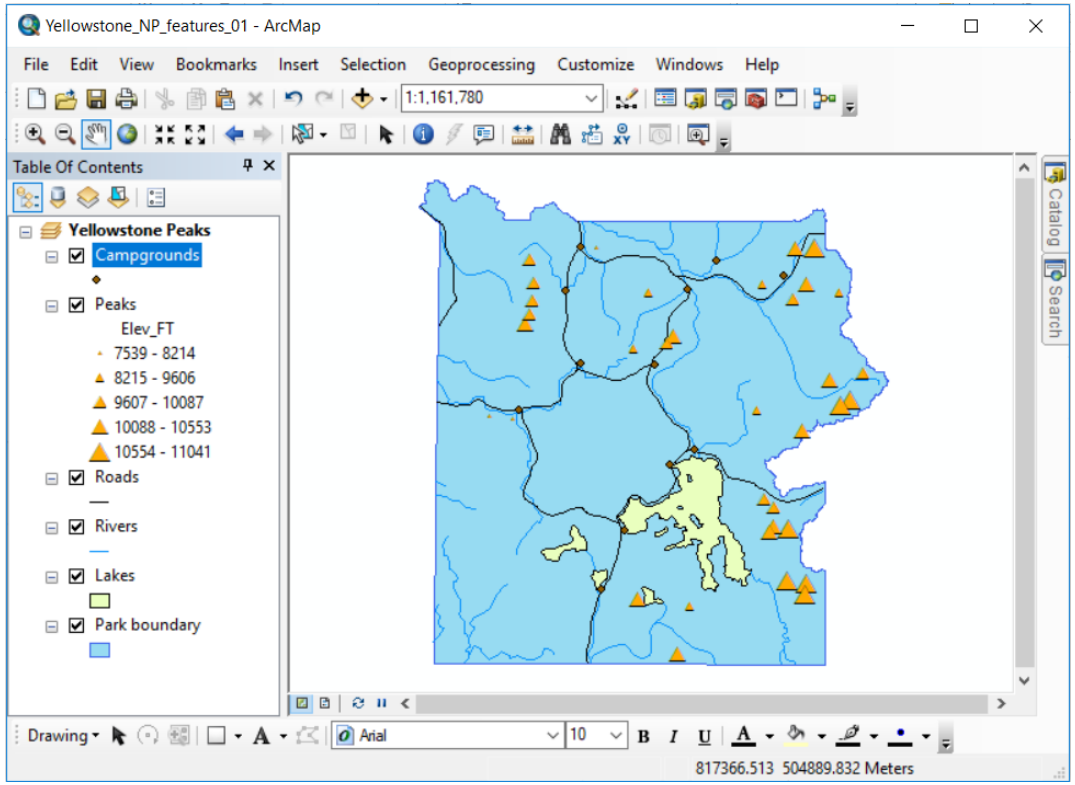
Übersicht

1. Logistik
2. Definitionen, Funktionen, Anwendungen
3. Koordinatensysteme und -transformationen
4. Räumliche Datenmodellierung
5. Vermaschungen
6. Räumliche Interpolation
7. Transformationen, Filtermethoden, Sonstiges

Was ist GIS

GIS = Geoinformationssystem: Rechnergestütztes System aus Hardware, Software und Daten

Raumbezogene Problemstellungen lassen sich modellieren und bearbeiten, Daten/Information können digital erfasst, organisiert, analysiert und dargestellt werden



Was ist GIS

GIS = Geoinformationssystem: Rechnergestütztes System aus Hardware, Software und Daten

Raumbezogene Problemstellungen lassen sich modellieren und bearbeiten, Daten/Information können digital erfasst, organisiert, analysiert und dargestellt werden

Vier-Komponenten-Modell (funktionelle Komponenten), IMAP

- Eingabe/Input (manuelle oder automatisierte Eingabe von Rohdaten)
- Management/Verwaltung (Aufbau von Datenbanken und Schnittstellen, sowie Bearbeitung von Geoobjekten)
- Analyse/Auswertung (Erstellung und Auswertung von Modellen zu gegebenen Sachverhalten)
- Präsentation

Was ist GIS

GIS = Geoinformationssystem: Rechnergestütztes System aus Hardware, Software und Daten

Raumbezogene Problemstellungen lassen sich modellieren und bearbeiten, Daten/Information können digital erfasst, organisiert, analysiert und dargestellt werden

Vier-Komponenten-Modell (funktionelle Komponenten), IMAP

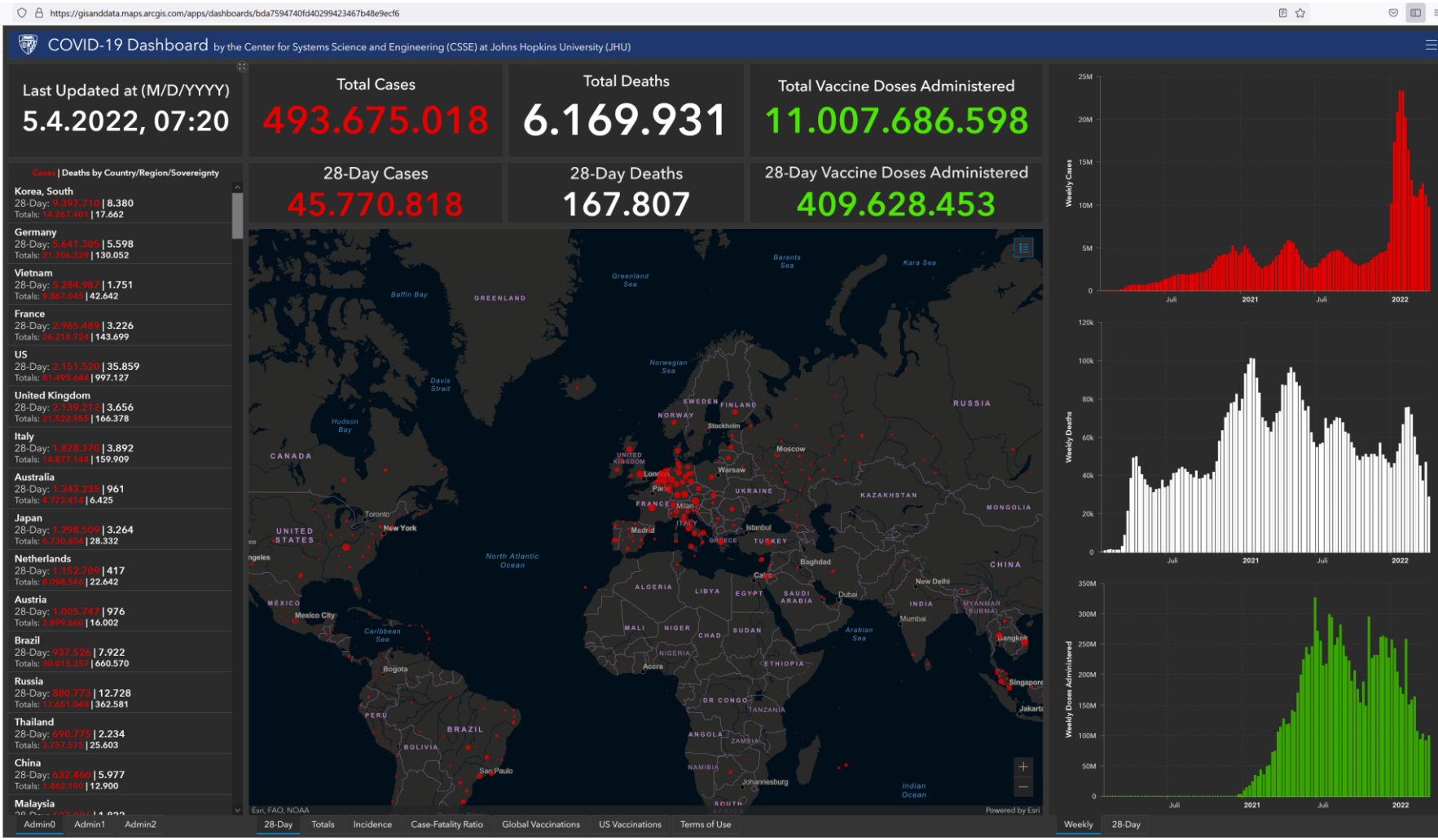
- Eingabe/Input (manuelle oder automatisierte Eingabe von Rohdaten)
- Management/Verwaltung (Aufbau von Datenbanken und Schnittstellen, sowie Bearbeitung von Geoobjekten)
- Analyse/Auswertung (Erstellung und Auswertung von Modellen zu gegebenen Sachverhalten)
- Präsentation

Strukturelle Komponenten

Aktuelles Beispiel: Corona



Was ist GIS


https://gisanddata.maps.arcgis.com/apps/dashboards/bda7594740fd40299423467b48e9ecf6
COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU)
Last Updated at (M/D/YYYY)
5.4.2022, 07:20
Total Cases
493.675.018
Total Deaths
6.169.931
Total Vaccine Doses Administered
11.007.686.598
28-Day Cases
45.770.818
28-Day Deaths
167.807
28-Day Vaccine Doses Administered
409.628.453
Cases | Deaths by Country/Region/Sovereignty
Korea, South
28-Day: 9.397.710 | 8.380
Totals: 14.267.401 | 17.662
Germany
28-Day: 5.641.305 | 5.598
Totals: 21.706.329 | 130.052
Vietnam
28-Day: 5.284.987 | 1.751
Totals: 9.867.045 | 42.642
France
28-Day: 2.965.489 | 3.226
Totals: 26.218.724 | 143.699
US
28-Day: 2.151.520 | 35.859
Totals: 81.495.644 | 997.127
United Kingdom
28-Day: 2.139.212 | 3.656
Totals: 21.522.955 | 166.378
Italy
28-Day: 1.828.370 | 3.892
Totals: 14.877.144 | 159.909
Australia
28-Day: 1.343.235 | 961
Totals: 4.772.414 | 6.425
Japan
28-Day: 1.298.509 | 3.264
Totals: 6.730.658 | 28.332
Netherlands
28-Day: 1.152.709 | 417
Totals: 8.098.546 | 22.642
Austria
28-Day: 1.005.747 | 976
Totals: 3.899.660 | 16.002
Brazil
28-Day: 937.526 | 7.922
Totals: 30.015.357 | 660.570
Russia
28-Day: 880.773 | 12.728
Totals: 17.651.048 | 362.581
Thailand
28-Day: 690.775 | 2.234
Totals: 3.757.575 | 25.603
China
28-Day: 632.460 | 5.977
Totals: 1.462.198 | 12.900
Malaysia
Admin0
Admin1
Admin2
28-Day
Totals
Incidence
Case-Fatality Ratio
Global Vaccinations
US Vaccinations
Terms of Use
Esri, FAO, NOAA
Powered by Esri
Weekly Cases
Weekly Deaths
Weekly Doses Administered
Juli
2021
2022
Weekly
28-Day

Daten- und Objektypen

- **Primärdaten**: direkt beobachtete oder gemessene Daten; Eingangsdaten
- **Sekundärdaten**: nicht direkt beobachtete Daten, aus Primärdaten durch Modellierung und Bearbeitung hervorgegangene Daten; Ausgangsdaten

Daten- und Objektypen

- **Primärdaten**: direkt beobachtete oder gemessene Daten; Eingangsdaten
- **Sekundärdaten**: nicht direkt beobachtete Daten, aus Primärdaten durch Modellierung und Bearbeitung hervorgegangene Daten; Ausgangsdaten

- **Klassische Daten** (Primär- und/oder Sekundärdaten)
- **Metadaten** (Daten über Daten; beschreibende Informationen): Zum Beispiel
  - Formate, Koordinatensysteme
  - Datenqualität, Zeitpunkt
  - Adressen, Vorwahlen
  - *Herkunftsdaten* (Provenance/Lineage): verwendete Methoden, Bearbeitungsschritte, Autoren, genutzte Primärdaten, etc.

Daten- und Objektypen

- **Primärdaten**: direkt beobachtete oder gemessene Daten; Eingangsdaten
- **Sekundärdaten**: nicht direkt beobachtete Daten, aus Primärdaten durch Modellierung und Bearbeitung hervorgegangene Daten; Ausgangsdaten

- **Klassische Daten** (Primär- und/oder Sekundärdaten)
- **Metadaten** (Daten über Daten; beschreibende Informationen): Zum Beispiel
  - Formate, Koordinatensysteme
  - Datenqualität, Zeitpunkt
  - Adressen, Vorwahlen
  - *Herkunftsdaten* (Provenance/Lineage): verwendete Methoden, Bearbeitungsschritte, Autoren, genutzte Primärdaten, etc.

- **Daten ≠ Information**: Information entsteht durch Anwendung von Regeln und Anweisungen auf Daten und liefert interpretierbare Ergebnisse; Information ist stets an ein Informations-/ Kommunikationsmittel gebunden (Sprache, Schrift, Visualisierung)
  - Syntax: interne Codierung, Struktur und Repräsentation der Information
  - Semantik: Kontext der zugrundeliegenden Daten
  - Kommunikation: Verwendung und Wiedergabe der zugrundeliegenden Daten

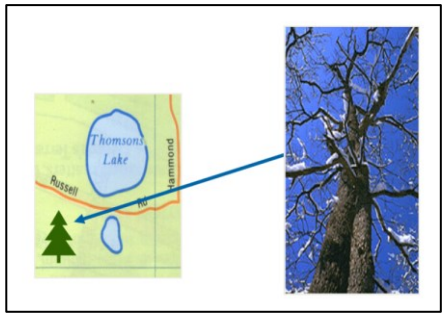
Daten- und Objektypen

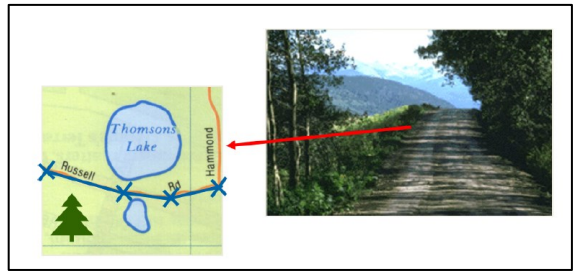
- **Geodaten vs. Sachdaten**:
  - Geodaten weisen stets einen Raumbezug auf und können in verschiedenen Datenmodellen (z.B. Vektor- oder Rasterdaten) vorliegen; es wird oft zwischen Geometriedaten und Topologiedaten unterschieden
  - Sachdaten liegen initial ohne Raumbezug vor (Namen, Nummern, Bezeichnungen, etc.)

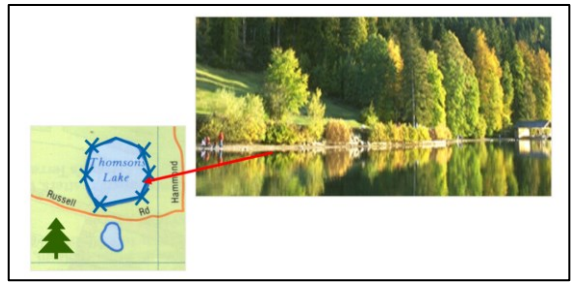
Daten- und Objektypen

- **Geodaten vs. Sachdaten**:
  - Geodaten weisen stets einen Raumbezug auf und können in verschiedenen Datenmodellen (z.B. Vektor- oder Rasterdaten) vorliegen; es wird oft zwischen Geometriedaten und Topologiedaten unterschieden
  - Sachdaten liegen initial ohne Raumbezug vor (Namen, Nummern, Bezeichnungen, etc.)

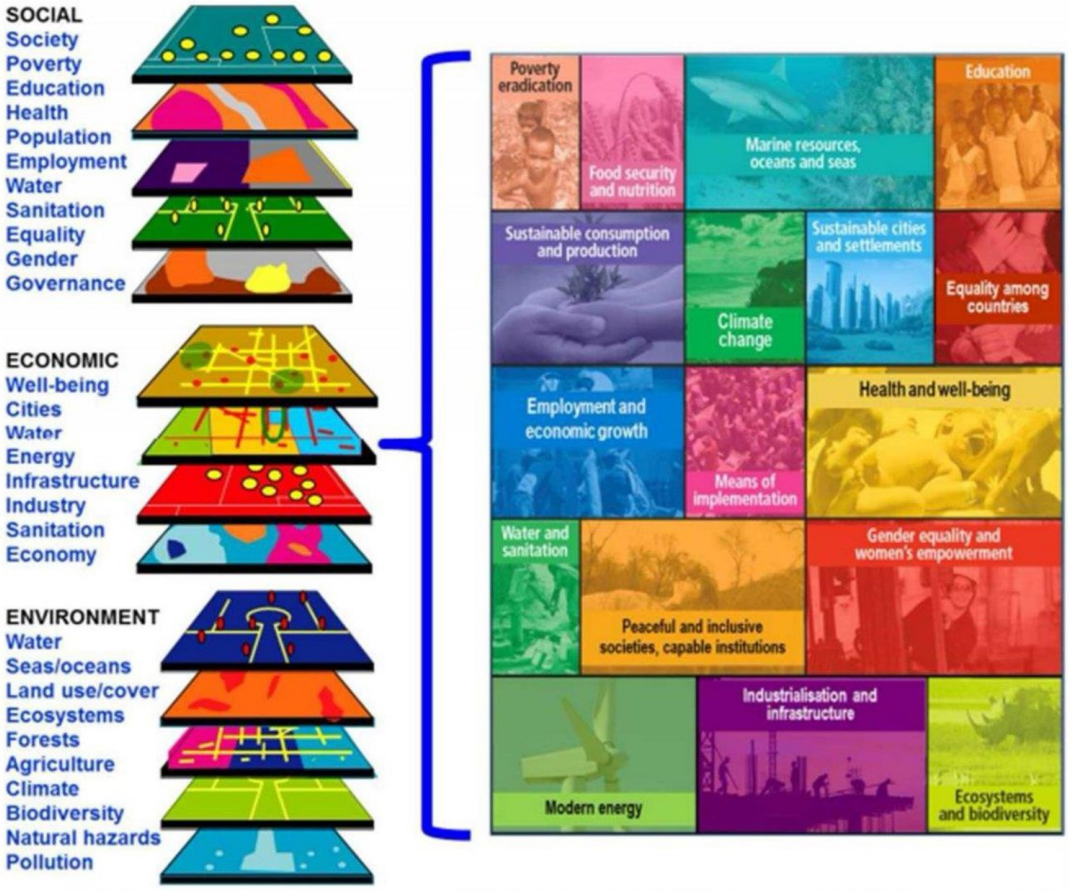
- **Geoobjekte**: Gruppierung von Daten in zusammenhängende, räumlich abgeschlossene Objekte mit ähnlichen Eigenschaften
  - Punkteobjekte (0-d)
  - Linienobjekte (1-d)
  - Flächenobjekte (2-d)
  - Volumenobjekte (3-d)
  - etc.





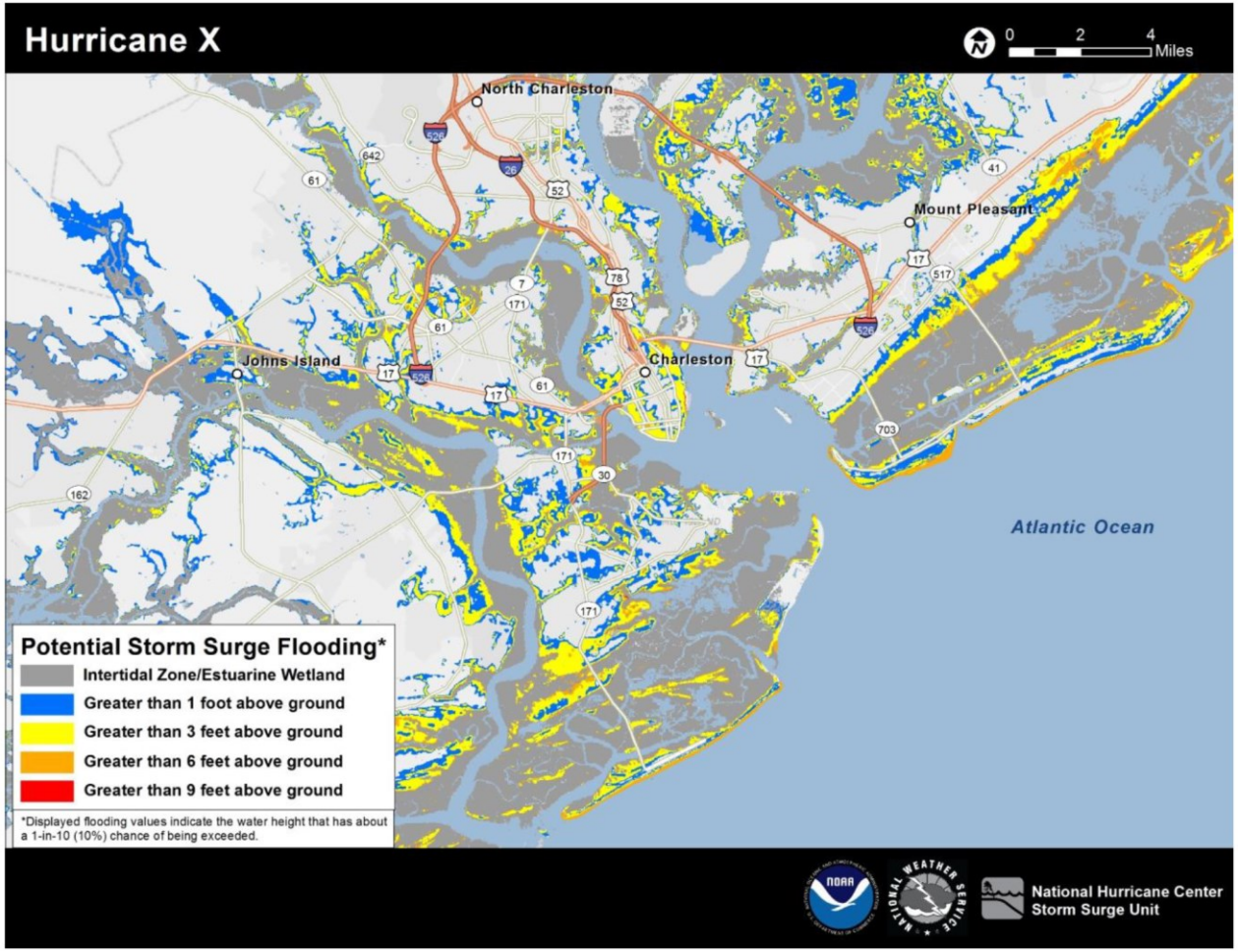


Daten- und Objektypen



Daten- und Objektypen

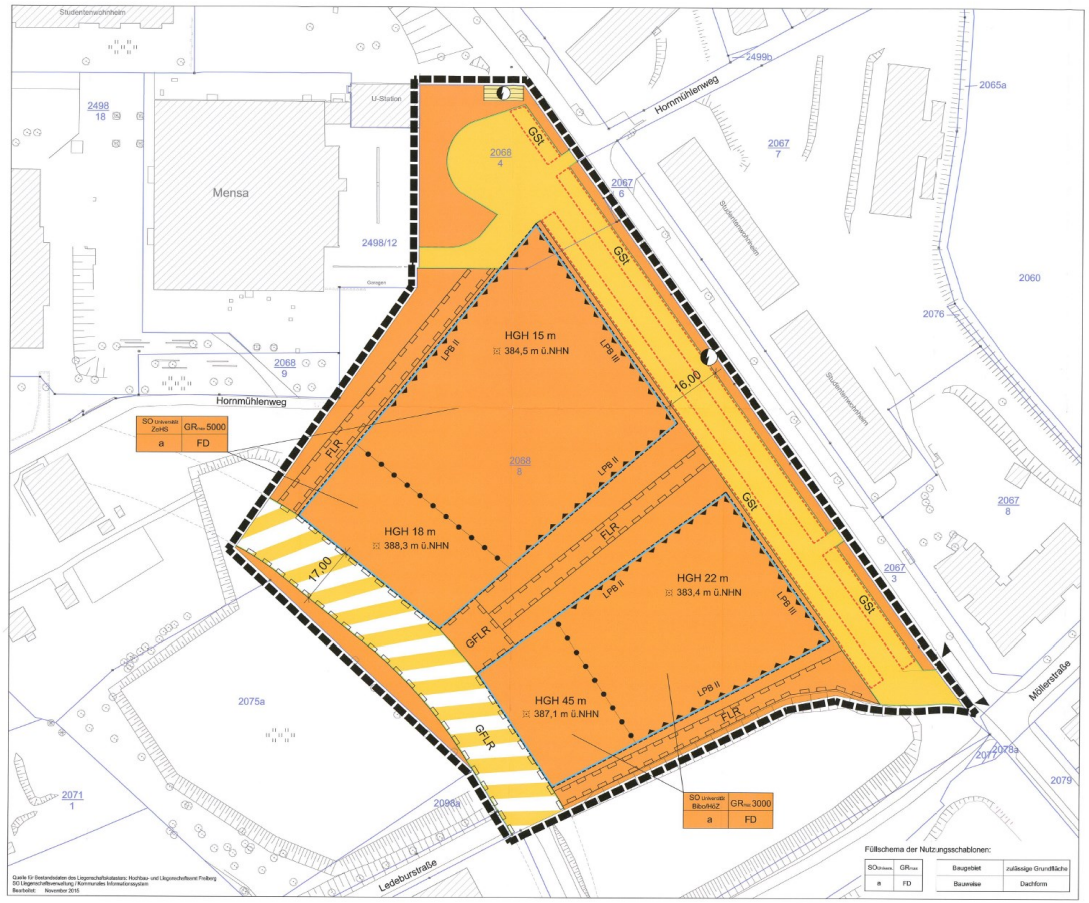
Primär- oder Sekundärdaten?



Daten- und Objektypen

Primär- oder Sekundärdaten?

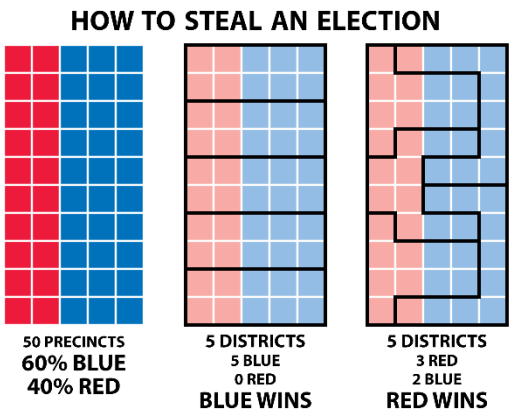
Vorhabenbezogener Bebauungsplan Nr. V 020 "Campuserweiterung am Wissenschaftskorridor"



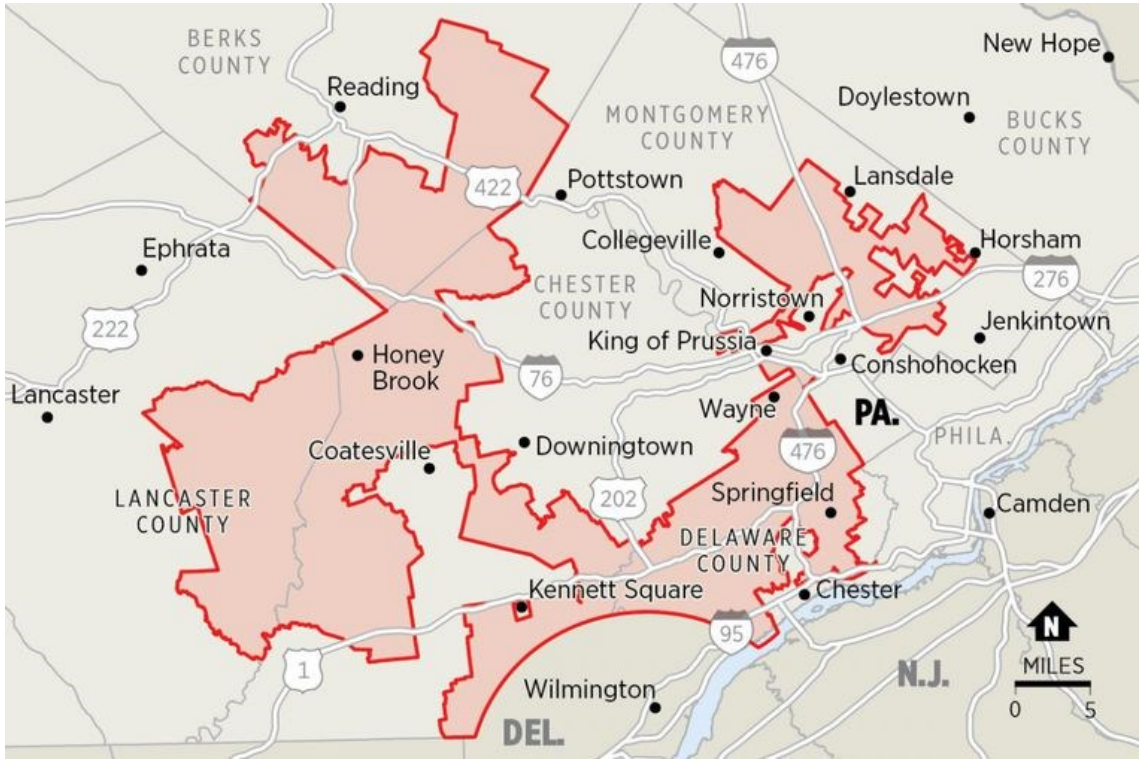


Daten- und Objektypen

Primär- oder Sekundärdaten?



https://www.fairvote.org/new_poll_everybody_hates_gerrymandering



https://www.inquirer.com/philly/opinion/pennsylvania-republicans-redistricting-gerrymandering-supreme-court-opinion-20180212.html

Funktionen eines GIS

Ein GIS Projekt unterteilt sich im Wesentlichen in drei Abschnitte:

- Aufbau der Datenbank
- Datenverarbeitung
- Datenintegration und -modellierung

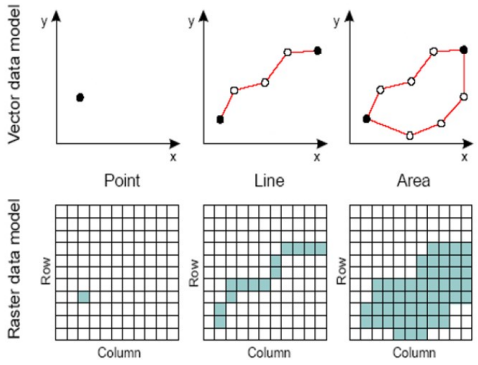
Funktionen eines GIS

Ein GIS Projekt unterteilt sich im Wesentlichen in drei Abschnitte:

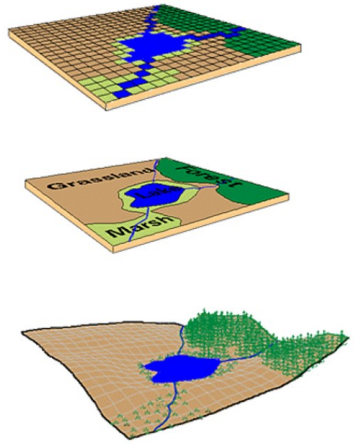
- Aufbau der Datenbank
- Datenverarbeitung
- Datenintegration und -modellierung

Grundsätzliche Funktionen eines GIS:

- **Dateneingabe und -aufnahme:**
  - Datenbank zu Hinterlegung von Daten unterschiedlichen Typs und Struktur
  - *Datenmodelle* (konzeptionelle Struktur; z.B. Raster- oder Vektormodell)
  - *Datenstruktur* (konkrete Umsetzung des Datenmodells)
  - *Datenformat* (konkretes Format zur Speicherung einer Datenstruktur)



- RASTER
- VECTOR
- Real World



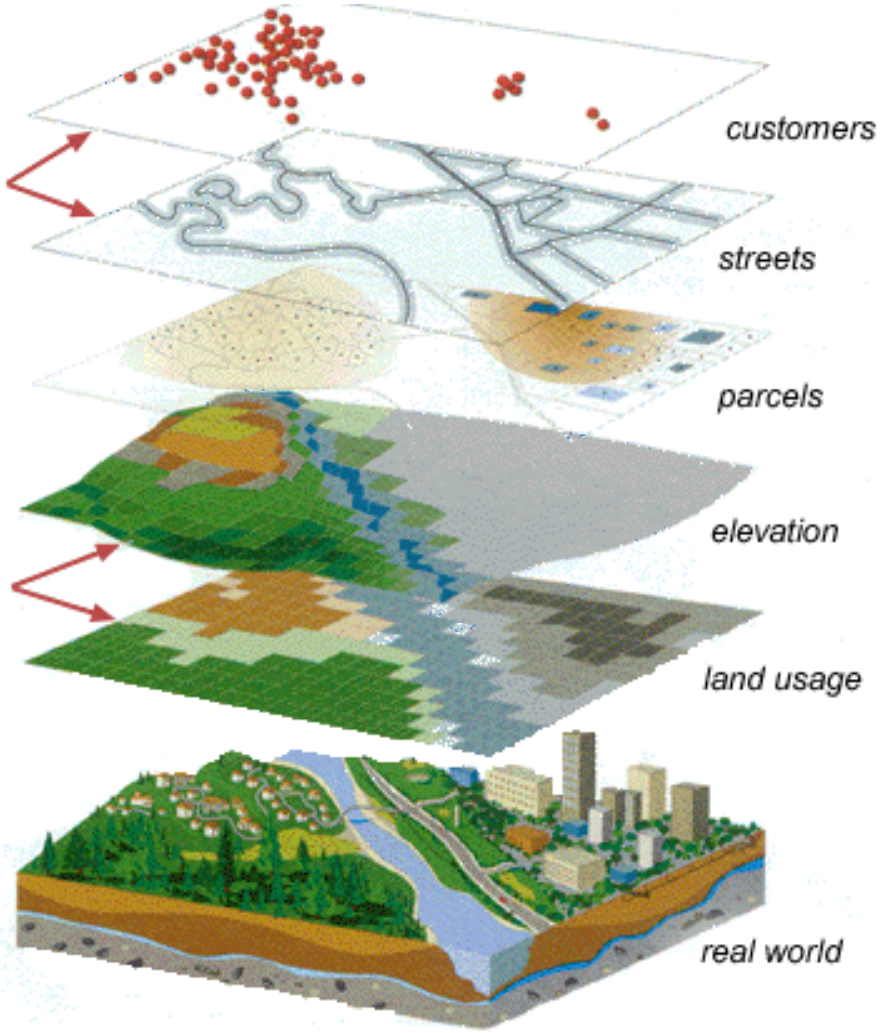
Funktionen eines GIS

Ein GIS Projekt u
- Aufbau der D
- Datenverarbe
- Datenintegrat

Grundsätzliche F
- **Dateneingab**
  - Datenb
  - *Datenm*
  - *Datens*
  - *Datenfo*
  - Datenla
    Ebener
    Reihent

v e c t o r

r a s t e r



nd Struktur
nodell)

ktur)
gruppiert;
ert werden;

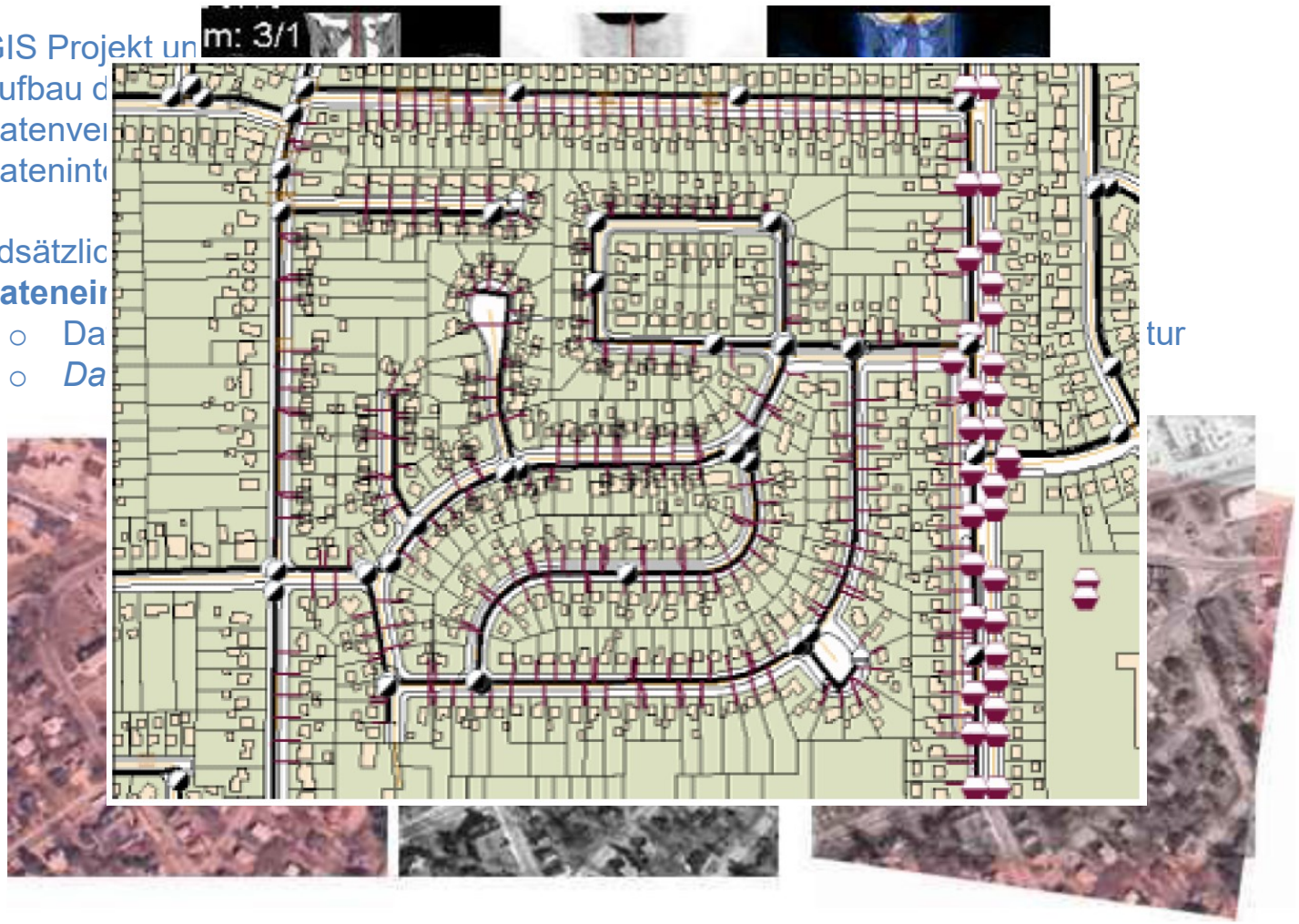
Funktionen eines GIS

Ein GIS Projekt un
- Aufbau d
- Datenver
- Datenint

Grundsätzlic
- **Dateneir**
  - Da
  - *Da*

tur

aten
g



Funktionen eines GIS

Ein GIS Projekt unterteilt sich im Wesentlichen in drei Abschnitte:

- Aufbau der Datenbank
- Datenverarbeitung
- Datenintegration und -modellierung

Grundsätzliche Funktionen eines GIS:

- **Dateneingabe und -aufnahme:**
  - Datenbank zu Hinterlegung von Daten unterschiedlichen Typs und Struktur
  - *Datenmodelle* (konzeptionelle Struktur; z.B. Raster- oder Vektormodell)
  - *Datenstruktur* (konkrete Umsetzung des Datenmodells)
  - Datenlayer/-ebenen (zugehörige Daten werden als Ebene/Layer gruppiert; Ebenen/Layer haben gleichen Raumbezug und können überlagert werden; Reihenfolge kann eine Rolle spielen)
  - *Georeferenzierung* und Registrierung (Zuweisung der direkten Raumbezüge der Daten in einen Raumbezug innerhalb des GIS (primäre Metrik); ggfs. auch Referenzierung von Sachdaten mit indirekten Raumbezug möglich, z.B. via PLZ, Wahlbezirk, etc. (sekundäre Metrik)
- **Visualisierung** (der Mensch kann komplexe räumliche Beziehungen visuell besser wahrnehmen als Texte, Tabellen, etc.; ein GIS sollte verschiedene Darstellungsmethoden umfassen)

Funktionen eines GIS

- **Räumlich**
  Visualisie
  - Wel
  - Wel
  - Top
    wen

:hen als bloße

tik auf?
n oder Objekten mit

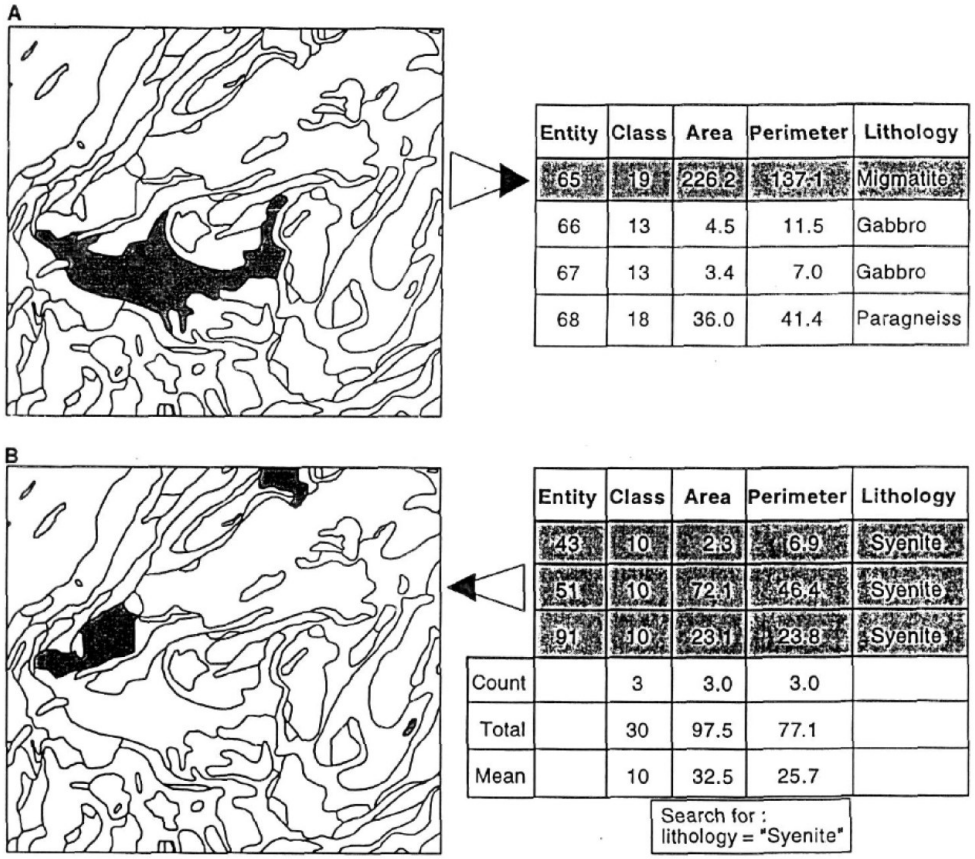


FIG. 5-14. Interactive spatial query of a geological map. A. Identifying the attributes of a polygon *selected on a map view*, as indicated in an associated polygon attribute table. B. Identifying those polygons on the map that have the attribute called lithology equal to "syenite", as *selected from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

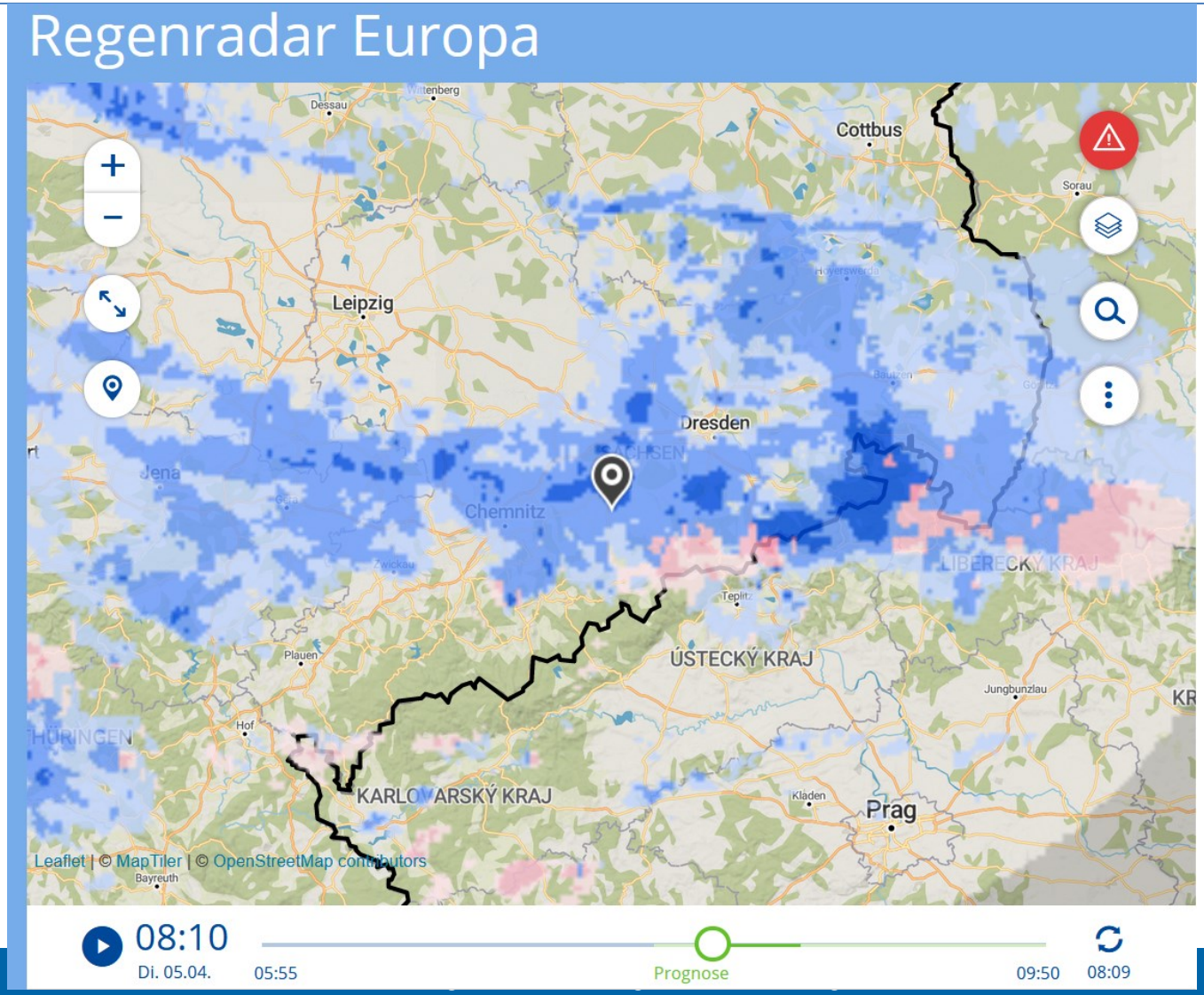
Räumliche Abfrage nach Bonham-Carter (1994)

Funktionen eines GIS

- **Räumliche Abfragen** (oft besser um spezielle Sachverhalte zu untersuchen als blosse Visualisierung):
  - Welche Charakteristik weist ein räumliche Position auf?
  - Welche räumlichen Positionen weisen eine bestimmte Charakteristik auf?
  - Topologische Abfragen (z.B. Abfragen nach benachbarten Objekten oder Objekten mit weniger als *x* km Entfernung)
- **Datenkombination/-integration**: Kombination verschiedener Datensätze zur gemeinsamen Interpretation
  - *Datenebene*: Nutzung mathematische Beziehungen und Modelle zur Erstellung neuer Attribute (z.B. Bestimmung des Verhältnisses zweier Elemente zueinander basierend auf dem Vorkommen jedes einzelnen)
  - *Layerebene*: Zusammenführung bereits existierender Layer zu neuem Layer mittels Map Algebra (z.B. kombiniere geologische Karten mit Luftbild, um bei verschiedenen Lithologien eine bestimmtes Spektralverhalten erkennen)
  - *Visualisierungsebene*: Überlagere verschiedene Karten zur gemeinsamen Visualisierung (bei Rasterdaten ist u.U. der Einsatz von Transparenzeigenschaften nötig)

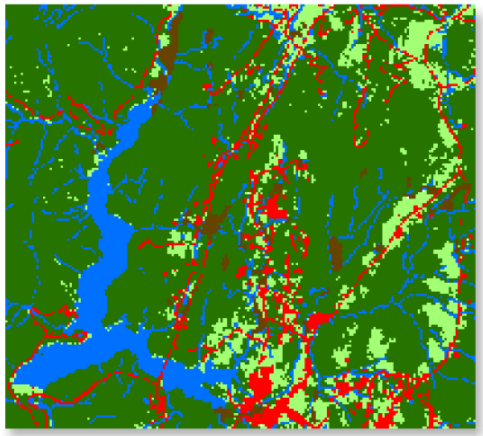
Beispiel: Regenradar

Regenradar


Regenradar Europa
Dessau
Cottbus
Sorau
Leipzig
Dresden
Jena
Chemnitz
Zwickau
Plauen
Hof
Teplitz
ÚSTECKÝ KRAJ
LIBERECKÝ KRAJ
Jungbunzlau
KARLOVARSKÝ KRAJ
Kladen
Prag
Bayreuth
Leaflet | © MapTiler | © OpenStreetMap contributors
08:10
Di. 05.04.
05:55
Prognose
09:50
08:09

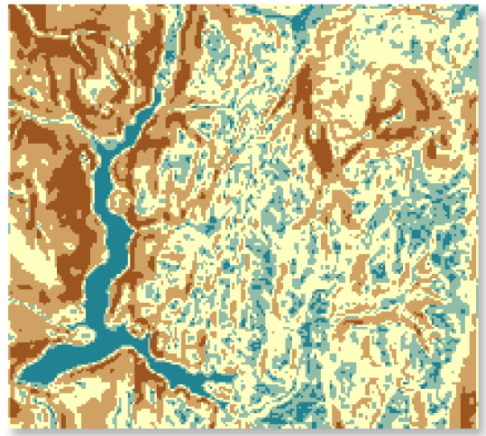
Funktionen eines GIS

- **R**
  Vi
- **D**
  In
- **D**
  (r



landu
LANDUSE
Agriculture
Built up
Forest
Water
Wetlands

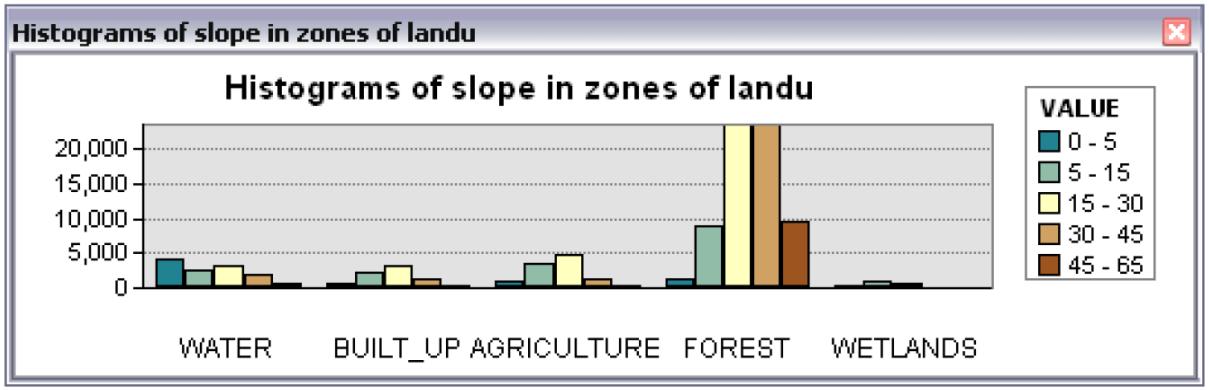
Zone input



slope
<VALUE>
0 - 5
5 - 15
15 - 30
30 - 45
45 - 65

Value input

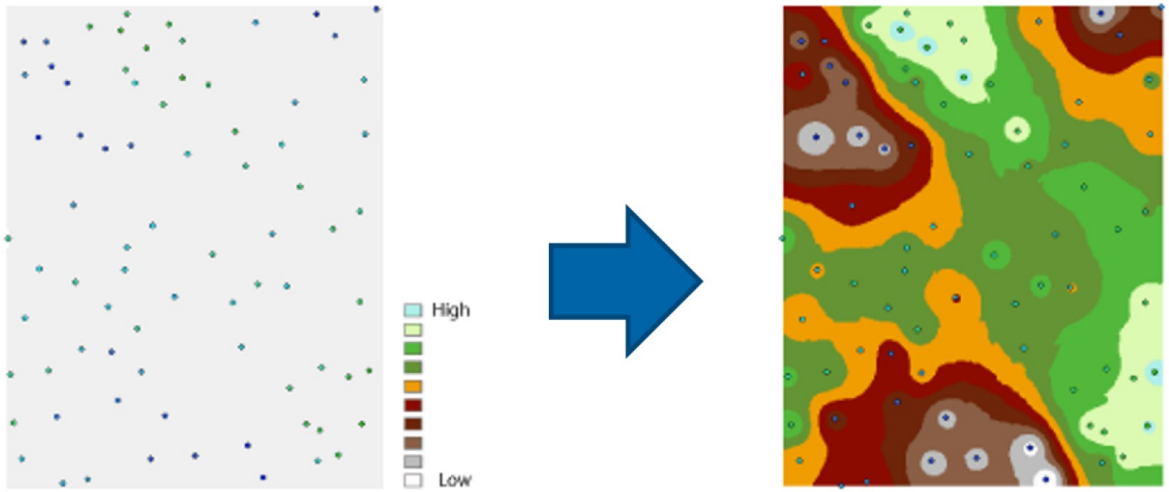
=



Output graph

Funktionen eines GIS

- **Vorhersage/Modellierung**: Vorhersage und Modellierung von Sachverhalten basierend auf gegebenen Daten (kombinierte mathematische Auswertung von Attributskombinationen zur Bestimmung eines unbekannten Attributs; oft in Verbindung mit Interpolation und Approximation)



Interpolation

Übersicht

1. Definitionen, Funktionen, Anwendungen (Vorlesung 1)
2. **Koordinatensysteme und -transformationen**
3. Räumliche Datenmodellierung
4. Vermaschungen
5. Räumliche Interpolation
6. Transformationen, Filtermethoden, Sonstiges

Raumbezug

Jedes Geoobjekt muss einen **Raumbezug** aufweisen. Die Zuweisung eines Raumbezuges durch einen Anwender zu einem Geoobjekt wird als **Georeferenzierung** (im engeren Sinne) bezeichnet.

Raumbezug

Jedes Geoobjekt muss einen **Raumbezug** aufweisen. Die Zuweisung eines Raumbezuges durch einen Anwender zu einem Geoobjekt wird als **Georeferenzierung** (im engeren Sinne) bezeichnet.

Direkter Raumbezug (primäre Metrik):

- Der Raumbezug über die Angabe zwei- oder dreidimensionalen Koordinaten hergestellt.
- Die Koordinaten beziehen sich auf ein bestimmtes Referenzsystem (*spatial reference*).
  - Lokal: Pixel-Koordinaten in Raster, Lokale Koordinaten relativ zu Referenzpunkt
  - Global: Koordinaten bzgl. Globalem Referenzsystem (UTM, Gauss-Krüger …)
- Distanzen zwischen Objekten können leicht berechnet werden.

Raumbezug

Jedes Geoobjekt muss einen **Raumbezug** aufweisen. Die Zuweisung eines Raumbezuges durch einen Anwender zu einem Geoobjekt wird als **Georeferenzierung** (im engeren Sinne) bezeichnet.

Direkter Raumbezug (primäre Metrik):

- Der Raumbezug über die Angabe zwei- oder dreidimensionalen Koordinaten hergestellt.
- Die Koordinaten beziehen sich auf ein bestimmtes Referenzsystem (*spatial reference*).
  - Lokal: Pixel-Koordinaten in Raster, Lokale Koordinaten relativ zu Referenzpunkt
  - Global: Koordinaten bzgl. Globalem Referenzsystem (UTM, Gauss-Krüger …)
- Distanzen zwischen Objekten können leicht berechnet werden.

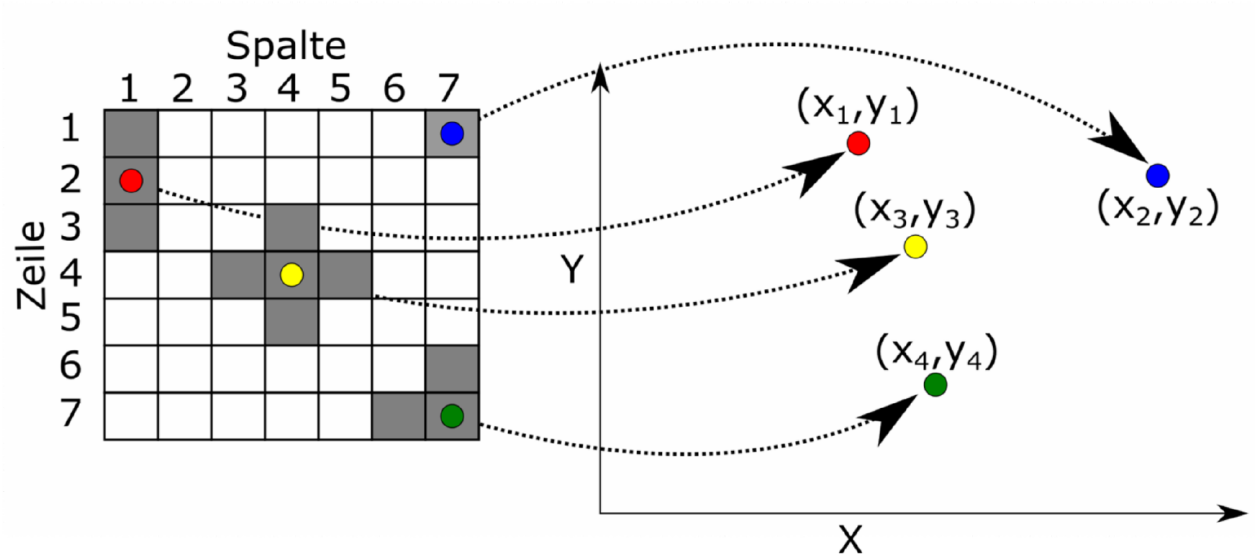
Indirekter Raumbezug (sekundäre Metrik):

- Raumbezug ist über qualitativen Größen angegeben, z.B.
  - Kennziffern (Postleitzahlen, Flurstücksnummern)
  - Namen von Orten / Gebieten
  - Adressen
  - …
- Distanzen zwischen Objekten können nur schwer berechnet werden
  - ➢ **Umwandlung in primäre Metrik!**

Georeferenzierung / Geokodierung

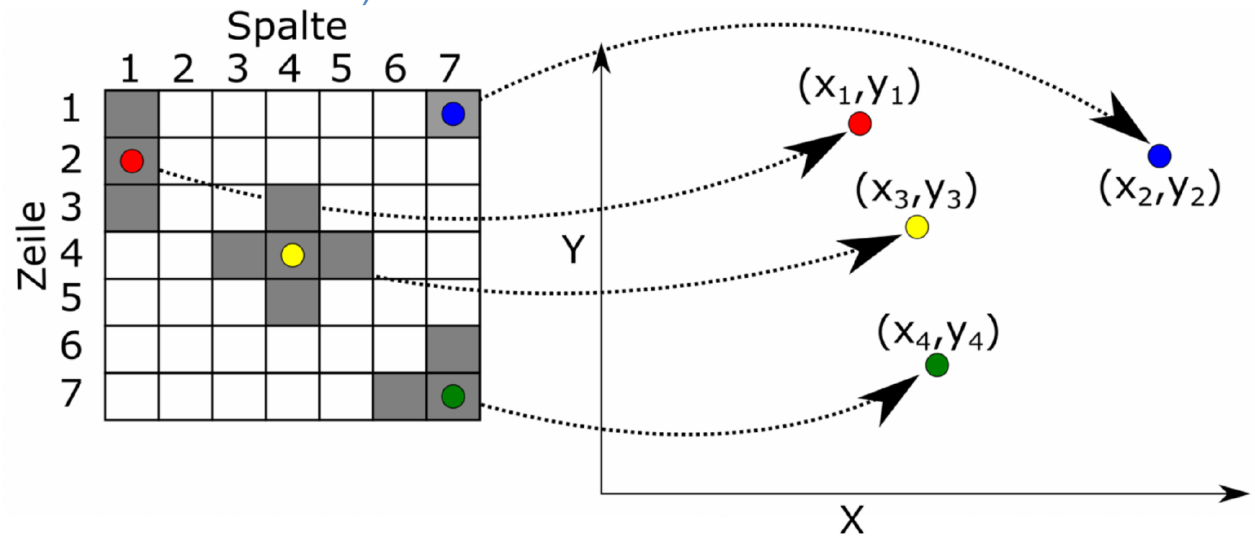
Im engeren Sinne bezeichnet **Georeferenzierung** die Zuweisung eines Raumbezuges zu einem Datensatz und **Geokodierung** die Überführung zwischen Koordinatensystemen.

Georeferenzierung umfasst (im weiteren Sinne; im ArcGIS-Kontext) den Workflow, welcher den externen Raumbezug der zu verwendenden Daten in den Raumbezug des GIS-Projektes umwandelt. Es muss der externe Raumbezug (z.B. geographische Koordinaten (λ,φ) oder Pixel in einem Luftbild) in ein üblicherweise durch (x,y)-Koordinaten gegebenen Raumbezug des GIS Projektes (lokales Koordinatensystem) überführt werden.



Georeferenzierung / Geokodierung

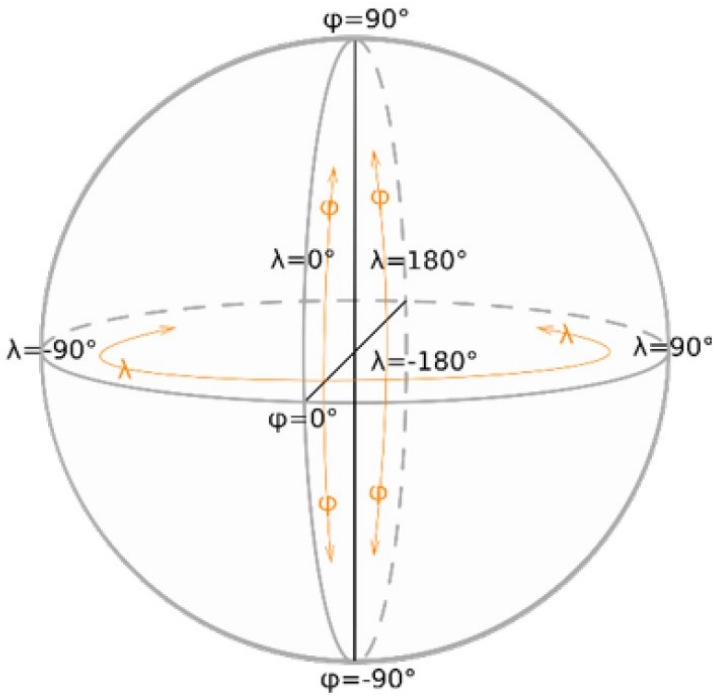
Eine der grundlegenden Aufgaben in einem GIS Projekt ist die Geokodierung, d.h. die Überführung des Raumbezuges aller Eingangsdatensätze in den gemeinsamen Raumbezug des GIS-Projektes. Dies ist notwendig, um verschiedene Datensätze räumlich miteinander vergleichen und kombinieren zu können (z.B. Luftbildaufnahme, Topographische Karte, geochemische Punktdaten).



Ein **erstes Problem**: Wie bekomme ich die kugelförmige Erde auf die Ebene projiziert?

Kartenprojektion

- Jeder Punkt ím Raum kann durch 3 kartesische Koordinaten x,y,z gegeben werden
- Problem 1: eine Karte ist üblicherweise 2-D
- Problem 2: die Erde ist annähernd kugelförmig



Transformation zwischen sphärischen Koordinaten (r, ϑ, φ) und kartesischen Koordinaten (x, y, z)

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r\cos(\lambda)sin(\varphi) \\ r\,sin(\lambda)\,sin(\varphi) \\ r\,cos(\varphi) \end{pmatrix}$$

für $r>0$, $0 \leq \lambda < 2\pi$, und $0 \leq \varphi \leq \pi$.

Umwandlung der Einheiten in “Grad”:
$-180° \leq \lambda < 180°$ und $-90° \leq \varphi \leq 90°$.

Um es 2D darstellen zu können brauchen wir eine Referenzfläche.

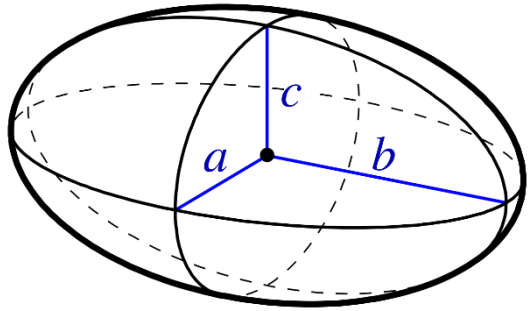
Kartenprojektion

Übliche Referenzflächen:

- Ellipsoid:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

- **Rotationsellipsoid**: $a = b; c$
  - Abplattung $f = \frac{a-c}{c}$
- Kugel: r=a=b=c (z.B. r=6371.2km)



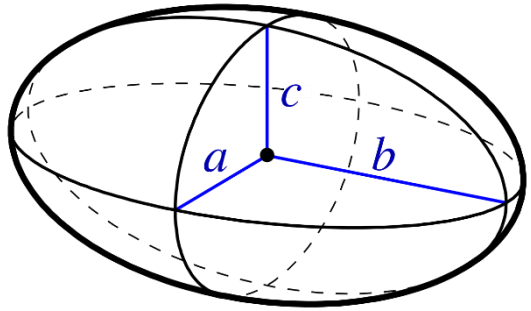
Kartenprojektion

Übliche Referenzflächen:

- Ellipsoid:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

- **Rotationsellipsoid**: $a = b; c$
  - Abplattung $f = \frac{a-c}{c}$
- Kugel: r=a=b=c (z.B. r=6371.2km)



Wichtige Referenzellipsoide sind z.B.

Name	a, b (in km)	c (in km)
WGS84 (1984, Nutzung für GPS)	6378.137	6356.752
Bessel (1841)	6377.379	6356.079
Hayford (1924)	6378.388	6356.912
Krassowskij (1940)	6378.245	6356.863

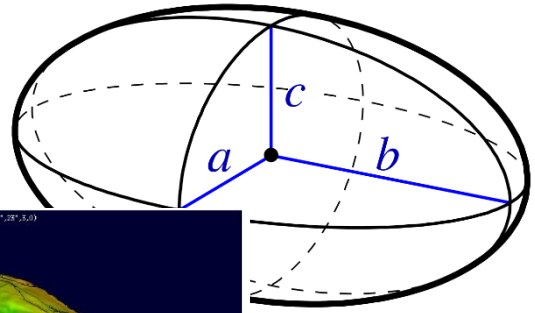
Kartenprojektion

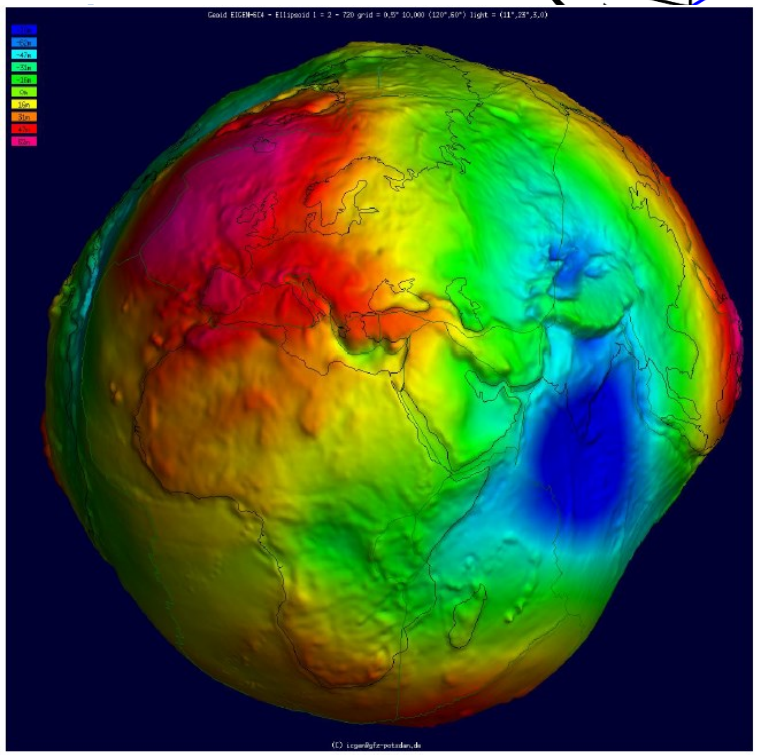
Übliche Referenzflächen:

- Ellipsoid:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

- **Rotationsellipsoid**:
  - Abplattung f =
- Kugel: r=a=b=c (z.B





Wichtige Referenzellips

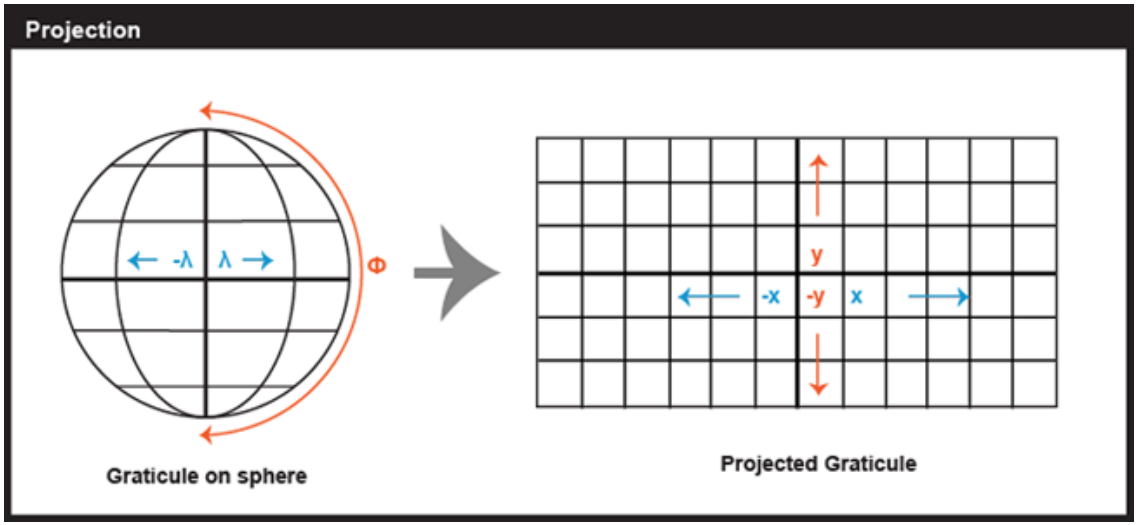
Name		c (in km)
WGS84 (1984, Nutzu		6356.752
Bessel (1841)		6356.079
Hayford (1924)		6356.912
Krassowskij (1940)		6356.863

- physikalische mot che eines konstanten Gravitationspotentials)

icgem.gfz-potsdam.de

Kartenprojektion

Es muss zur Darstellung in 2D jedoch die Oberfläche des Referenzkörpers (hier Kugel) auf eine Ebene projeziert werden (Höhe r wird meist als Attribut betrachtet): Ist eine bloße Darstellung mittels Längen- und Breitengraden in einem rechwinkligen Koordinatensystem sinnvoll?



www.e-education.psu.edu/geog160/node/1918

Problem: Finde eine geeignete Abbildung

$T : \mathbb{R}^3 \supset \mathbb{S}_r \to \mathbb{R}^2,$

oder alternativ $T : [0, 2\pi) \times [0, \pi] \to \mathbb{R}^2,$

Kartenprojektion

Es muss zur Darstellung in 2D jedoch die Oberfläche des Referenzkörpers (hier Kugel) auf eine Ebene projeziert werden (Höhe r wird meist als Attribut betrachtet): Ist eine bloße Darstellung mittels Längen- und Breitengraden in einem rechwinkligen Koordinatensystem sinnvoll?

Eine globale verzerrungsfreie Darstellung der Erde ist nicht möglich:

- Winkeltreu
- Längentreu
- Flächentreu

Kartenprojektion

Es muss zur Darstellung in 2D jedoch die Oberfläche des Referenzkörpers (hier Kugel) auf eine Ebene projeziert werden (Höhe r wird meist als Attribut betrachtet): Ist eine bloße Darstellung mittels Längen- und Breitengraden in einem rechwinkligen Koordinatensystem sinnvoll?

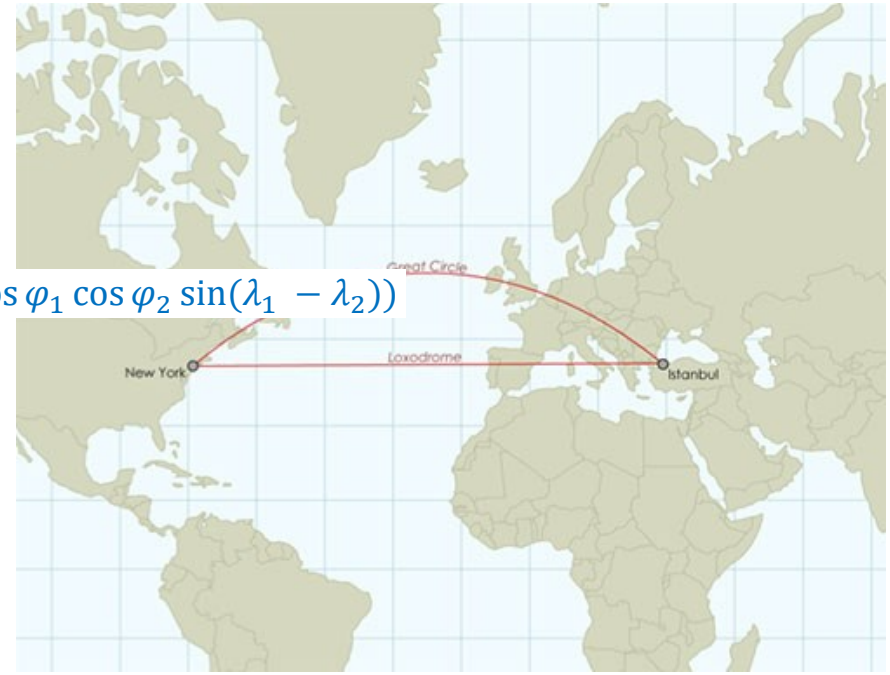
Eine globale verzerrungsfreie Darstellung der Erde ist nicht möglich:

- Winkeltreu
- **Längentreu**
- Flächentreu

Kürzeste Distanz auf kugelförmiger Erde:

$$\text{dist}_{sph}(P_1, P_2) = r\arcsin(\sin\varphi_1 \sin\varphi_2 + \cos\varphi_1 \cos\varphi_2 \sin(\lambda_1 - \lambda_2))$$

immer entlang eines Großkreises



https://www.axismaps.com/guide/map-projections

Kartenprojektion

Es muss zur Darstellung in 2D jedoch die Oberfläche des Referenzkörpers (hier Kugel) auf eine Ebene projeziert werden (Höhe r wird meist als Attribut betrachtet): Ist eine blosse Darstellung mittels Längen- und Breitengraden in einem rechwinkligen Koordinatensystem sinnvoll?

Eine globale verzerrungsfreie Darstellung der Erde ist nicht möglich:

- Winkeltreu
- **Längentreu**
- Flächentreu

Kürzeste Distanz auf kugelförmiger Erde:

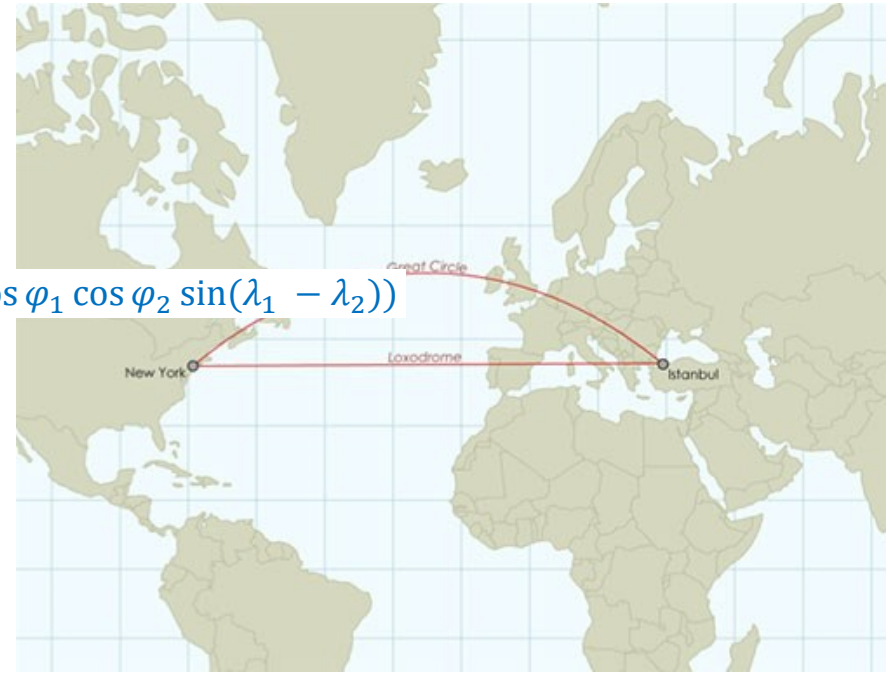
$$\text{dist}_{sph}(P_1, P_2) = r\arcsin(\sin\varphi_1 \sin\varphi_2 + \cos\varphi_1 \cos\varphi_2 \sin(\lambda_1 - \lambda_2))$$

immer entlang eines Großkreises

Kürzeste Distanz auf euklidischer Ebene:

$$\text{dist}_{eucl}(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

(Satz des Phytagoras)



https://www.axismaps.com/guide/map-projections

Kartenprojektion

Es muss zur Darstellung in 2D jedoch die Oberfläche des Referenzkörpers (hier Kugel) auf eine Ebene projeziert werden (Höhe r wird meist als Attribut betrachtet): Ist eine blosse Darstellung mittels Längen- und Breitengraden in einem rechwinkligen Koordinatensystem sinnvoll?

Eine globale verzerrungsfreie Darstellung der Erde ist nicht möglich:

- Winkeltreu
- **Längentreu**
- Flächentreu

Eine Projektion heißt *längentreu* (mit Maßstab α), falls

$$\mathrm{dist}_{eucl}(P_1, P_2) = \alpha \mathrm{dist}_{sph}(P_1, P_2)$$

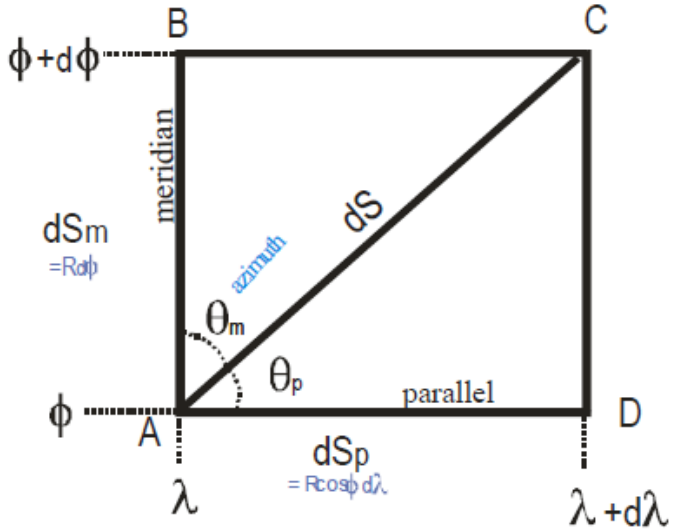
Global längentreue Abbildungen **gibt es nicht**, aber Längentreue unter bestimmten Annahmen kann garantiert warden (z.B. von einem zentralen Punkt aus)



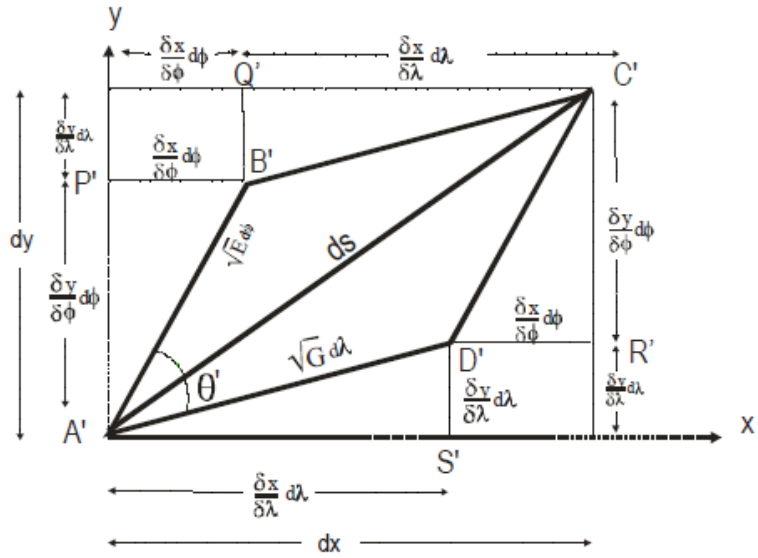
https://www.axismaps.com/guide/map-projections

Kartenprojektion

Infinitesimales Rechteck auf der Kugeloberfläche



Infinitesimales Rechteck in der Kartenebene



Beziehungen zwischen verschiedenen Kenngrössen:

$$(ds)^2 = (dx)^2 + (dy)^2 = \left(\frac{\partial x}{\partial \varphi}d\varphi + \frac{\partial x}{\partial \lambda}d\lambda\right)^2 + \left(\frac{\partial y}{\partial \varphi}d\varphi + \frac{\partial y}{\partial \lambda}d\lambda\right)^2$$

$$(ds)^2 = E(d\varphi)^2 + 2F(d\varphi\, d\lambda) + G(d\lambda)^2$$

E,F,G: Gaussche Fundamentalgrössen

Kartenprojektion

Beziehungen zwischen verschiedenen Kenngrössen:

$$(ds)^2 = (dx)^2 + (dy)^2 = \left(\frac{\partial x}{\partial \varphi} d\varphi + \frac{\partial x}{\partial \lambda} d\lambda\right)^2 + \left(\frac{\partial y}{\partial \varphi} d\varphi + \frac{\partial y}{\partial \lambda} d\lambda\right)^2$$

$$(ds)^2 = E(d\varphi)^2 + 2F(d\varphi\, d\lambda) + G(d\lambda)^2$$

E,F,G: Gaussche Fundamentalgrössen

Mit

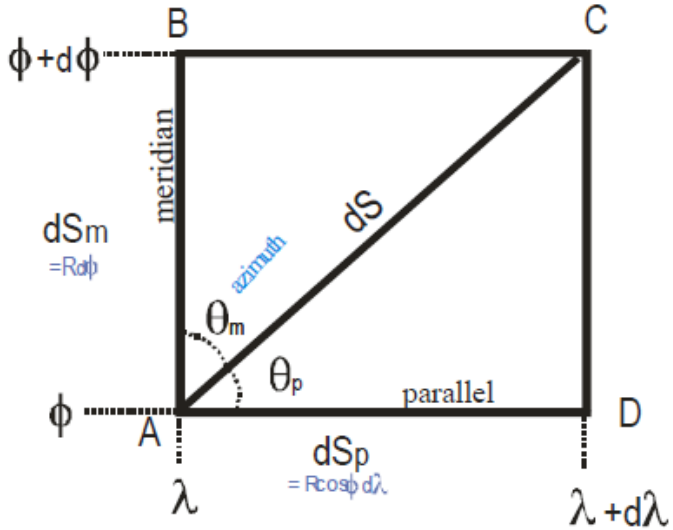
$$E(\,d\varphi\,)^2 = \left(\left(\frac{\partial x(\lambda,\varphi)}{\partial \varphi}\right)^2 + \left(\frac{\partial y(\lambda,\varphi)}{\partial \varphi}\right)^2\right) d\varphi^2$$

$$G(\,d\lambda\,)^2 = \left(\left(\frac{\partial x(\lambda,\varphi)}{\partial \lambda}\right)^2 + \left(\frac{\partial y(\lambda,\varphi)}{\partial \lambda}\right)^2\right) d\lambda^2$$

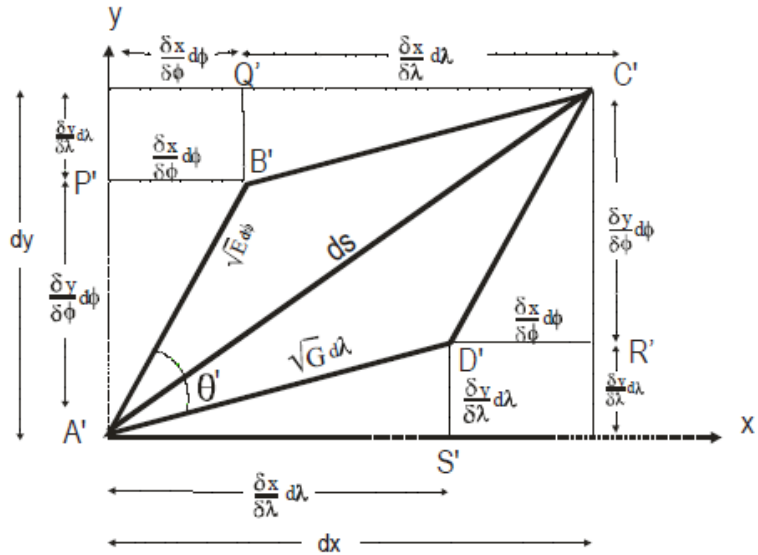
$$F(\,d\varphi d\lambda\,) = \left(\frac{\partial x(\lambda,\varphi)^2}{\partial \varphi \partial \lambda} + \frac{\partial y(\lambda,\varphi)^2}{\partial \varphi \partial \lambda}\right) d\varphi d\lambda$$

Kartenprojektion

Infinitesimales Rechteck auf der Kugeloberfläche



Infinitesimales Rechteck in der Kartenebene



Skalen/Massstäbe:

$$\alpha_\lambda = \frac{A'D'}{AD} = \frac{\sqrt{G}}{r\cos(\varphi)}$$

$$\alpha_\varphi = \frac{A'B'}{AB} = \frac{\sqrt{E}}{r}$$

$$\alpha = \frac{A'C'}{AC} = \frac{\sqrt{E(d\varphi)^2+2F(d\varphi\,d\lambda)+G(d\lambda)^2}}{\sqrt{(d\varphi)^2+(\cos\varphi)^2(d\lambda)^2}}$$

Kartenprojektion

Es muss zur Darstellung in 2D jedoch die Oberfläche des Referenzkörpers (hier Kugel) auf eine Ebene projeziert werden (Höhe r wird meist als Attribut betrachtet): Ist eine bloße Darstellung mittels Längen- und Breitengraden sinnvoll in einem rechwinkligen Koordinatensystem sinnvoll?

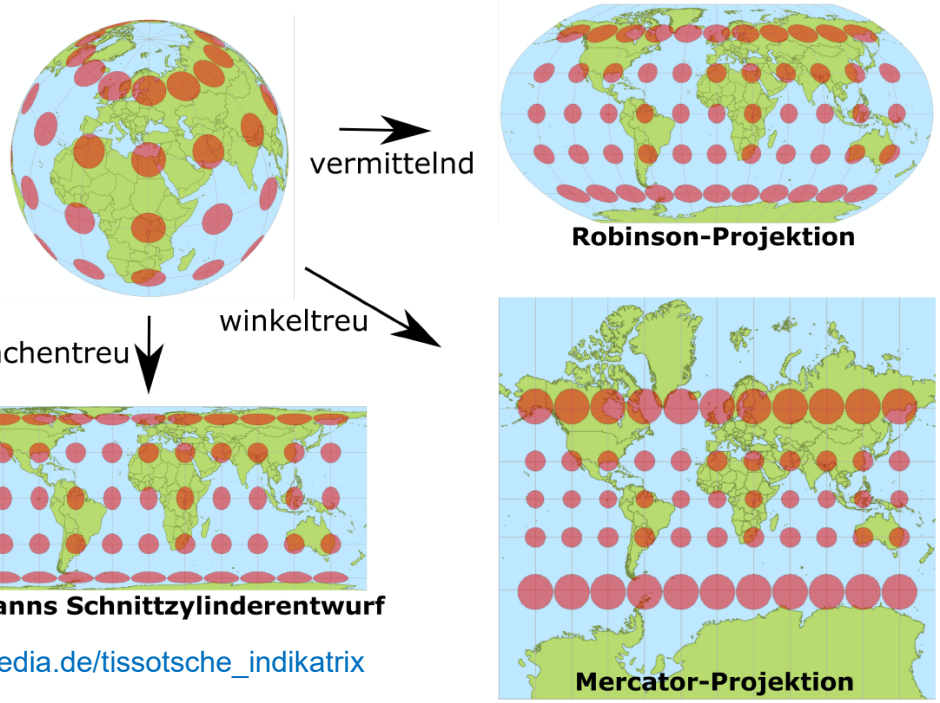
Eine globale verzerrungsfreie Darstellung der Erde ist nicht möglich:

- **Winkeltreu**
- Längentreu
- Flächentreu

Eine Abbildung $T : \mathbb{R}^3 \to \mathbb{R}^2$ heisst winkeltreu (konform), falls

$$\frac{\langle TP,TQ\rangle}{\|TP\|\|TQ\|} = \frac{\langle P,Q\rangle}{\|P\|\|Q\|}$$

für alle P, Q aus dem Tangentialraum der Sphäre, und falls die Determinante von T positiv ist.



wikipedia.de/tissotsche_indikatrix

Kartenprojektion

Es muss zur Darstellung in 2D jedoch die Oberfläche des Referenzkörpers (hier Kugel) auf eine Ebene projeziert werden (Höhe r wird meist als Attribut betrachtet): Ist eine bloße Darstellung mittels Längen- und Breitengraden sinnvoll in einem rechwinkligen Koordinatensystem sinnvoll?

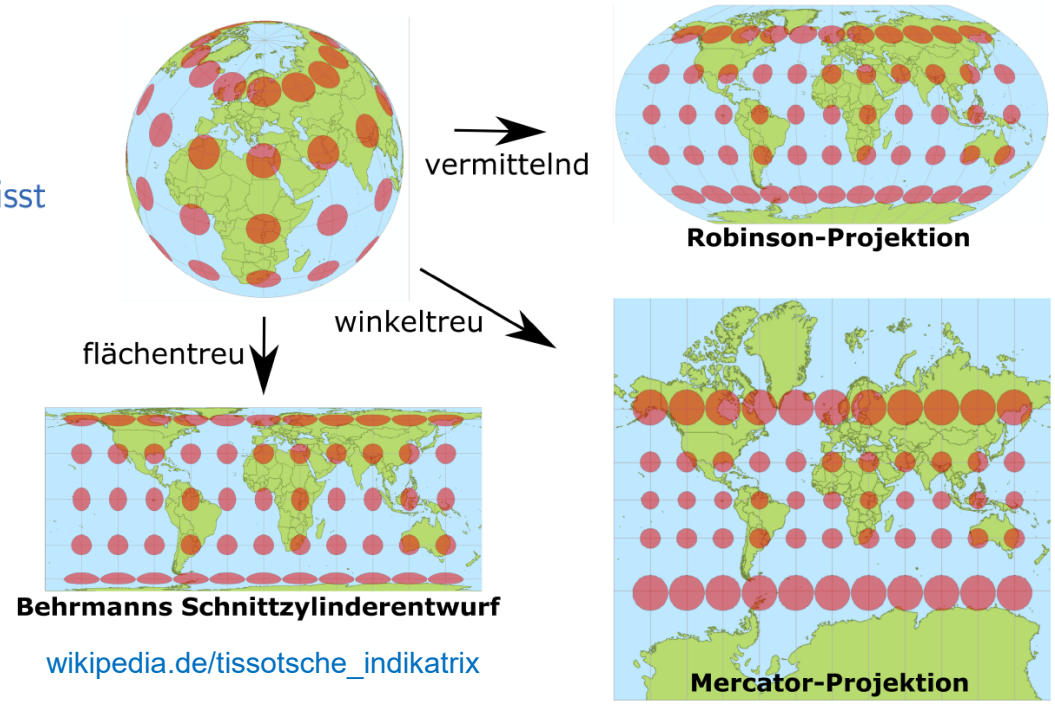
Eine globale verzerrungsfreie Darstellung der Erde ist nicht möglich:

- Winkeltreu
- Längentreu
- **Flächentreu**

Eine Abbildung $T : \mathbb{R}^3 \to \mathbb{R}^2$ heisst flächentreu, falls

$$\int_\Omega 1 \, d\mu = \alpha \int_{T(\Omega)} 1 \, d(x, y)$$

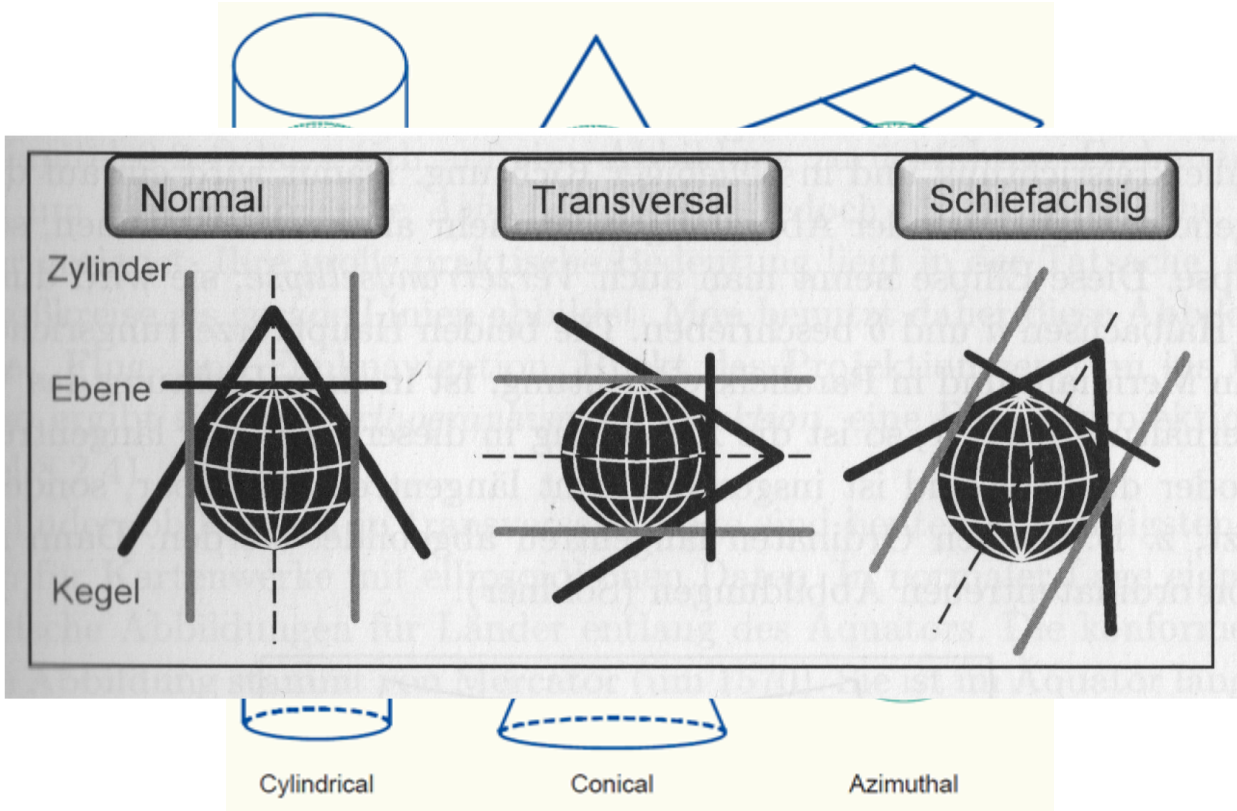
für jedes Gebiet $\Omega \subset \mathbb{S}_r$. Die Konstante α spiegelt dabei den Massstab wieder (Massstab sei hier als ein Skalierungsfaktor aufgefasst, nicht der umgangssprachliche Massstab).



wikipedia.de/tissotsche_indikatrix

Kartenprojektion

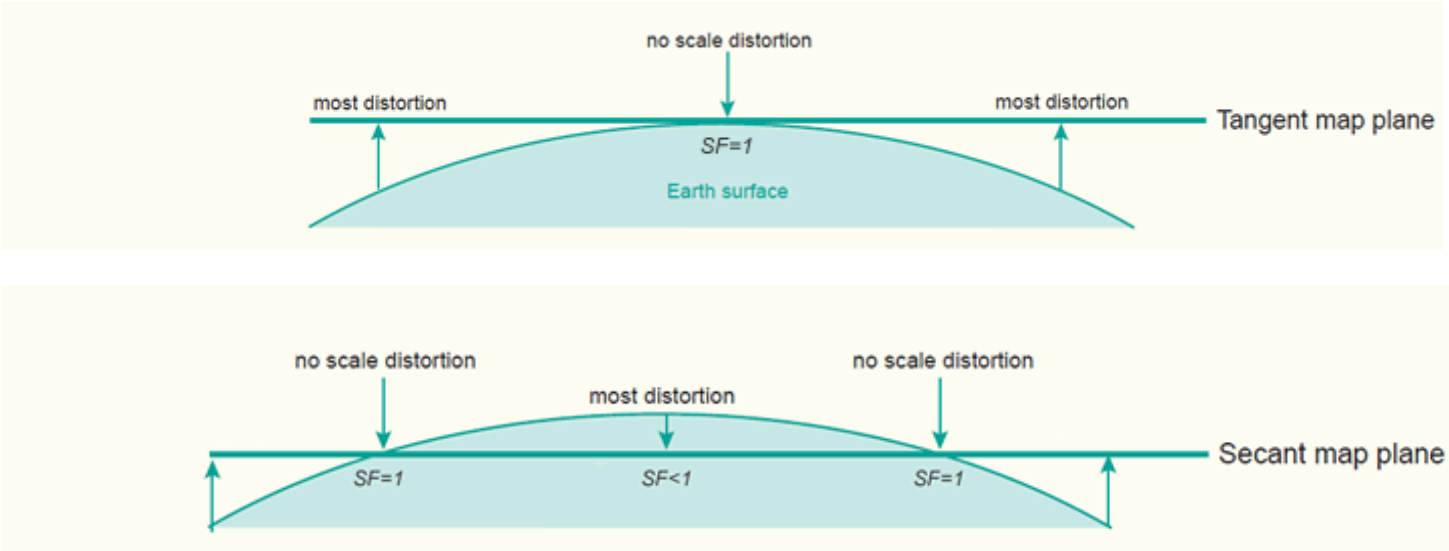
Man unterscheidet Kartenprojektionen basierend auf den Projektionsflächen



https://kartoweb.itc.nl

Kartenprojektion

Der Skalierungsfaktor beschreibt den Quotienten zwischen dem wahren Abstand zweier infinitesimal entfernten Punkte P und Q und dem Abstand der auf die Ebene projizierten Punkte TP und TQ.



https://kartoweb.itc.nl

Welche Projektionsebene würden Sie bevorzugen?

Kartenprojektion

Beispiele für winkeltreue Projektionen:

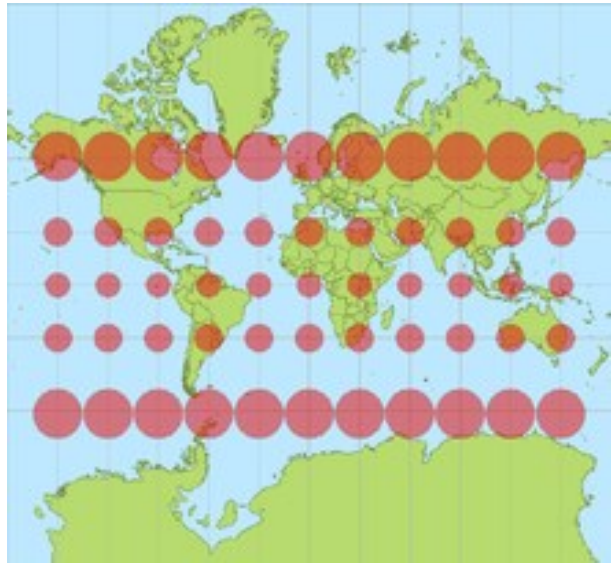
- **Mercator Projektion** (Zylinder-Projektion)

  Es sei $(x, y) = TP = T(\lambda, \varphi)$. Mit λ_0 dem Referenzmeridian (üblicherweise Greenwich $\lambda_0 = 0$):

  $$x = r(\lambda - \lambda_0)$$
  $$y = r \ln\left(\tan\left(\frac{\pi}{4} + \frac{\varphi}{2}\right)\right)$$

  Für die Inversion $(\lambda, \varphi) = T^{-1}(x, y)$ gilt

  $$\lambda = \lambda_0 + \frac{x}{r}$$
  $$\varphi = 2 \arctan\left(\exp\left(\frac{y}{r}\right)\right) - \frac{\pi}{2}$$



Kartenprojektion

Beispiele für winkeltreue Projektionen:

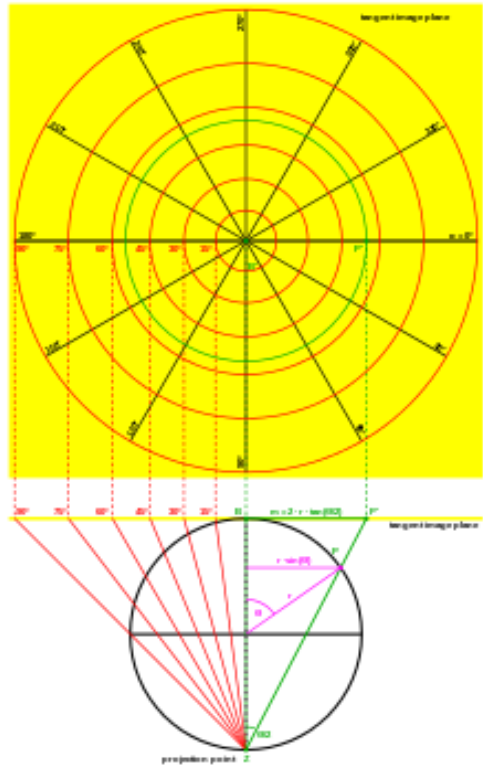
- **Mercator Projektion** (Zylinder-Projektion)

  Es sei $(x, y) = TP = T(\lambda, \varphi)$. Mit λ_0 dem Referenzmeridian (üblicherweise Greenwich $\lambda_0 = 0$):

  $$x = r(\lambda - \lambda_0)$$
  $$y = r \ln\left(\tan\left(\frac{\pi}{4} + \frac{\varphi}{2}\right)\right)$$

  Für die Inversion $(\lambda, \varphi) = T^{-1}(x, y)$ gilt

  $$\lambda = \lambda_0 + \frac{x}{r}$$
  $$\varphi = 2 \arctan\left(\exp\left(\frac{y}{r}\right)\right) - \frac{\pi}{2}$$



- **Stereographische Projektion** (azimuthale Projektion)

  Es sei $(x, y) = TP = T(\lambda, \varphi) = s(\cos(\alpha), \sin(\alpha))$, wobei

  $$\alpha = \lambda$$
  $$s = 2r \tan\left(\frac{\varphi}{2}\right)$$

Kartenprojektion

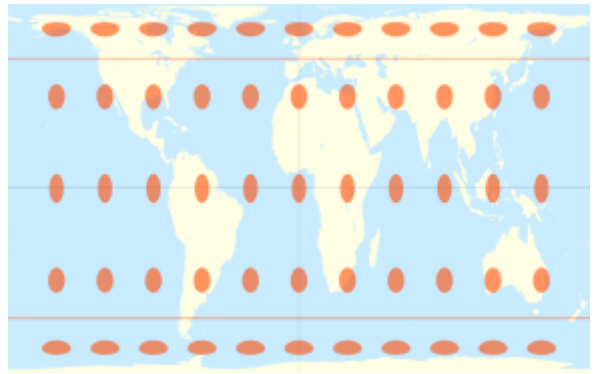
Beispiele für flächentreue Projektionen:

https://map-projections.net/compare.php

- **Gall-Peters Projektion** (Zylinder-Projektion)

  Es sei $(x, y) = TP = T(\lambda, \varphi)$:

  $$x = r\lambda$$
  $$y = 2r \sin(\varphi)$$



Kartenprojektion

Beispiele für flächentreue Projektionen:

https://map-projections.net/compare.php

- **Gall-Peters Projektion** (Zylinder-Projektion)

  Es sei $(x, y) = TP = T(\lambda, \varphi)$:

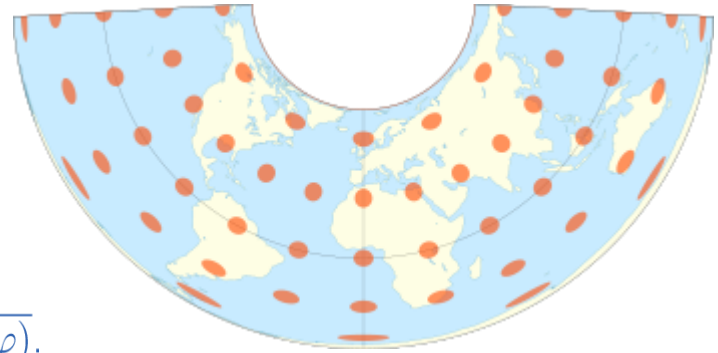
  $$x = r\lambda$$
  $$y = 2r \sin(\varphi)$$

- **Albers Kegelprojektion**

  Es sei $(x, y) = TP = T(\lambda, \varphi)$, sowie $\lambda_0, \varphi_1, \varphi_2$ Referenzlängen- bzw. -breitengrade:

  $$x = \rho \sin(\theta)$$
  $$y = \rho_0 - \rho \cos(\theta),$$

  wobei $\theta = \beta(\lambda - \lambda_0)$, $\beta = \frac{1}{2}(\sin(\varphi_1) + \sin(\varphi_2))$, $\gamma = \cos^2(\varphi_1) + 2\beta \sin(\varphi_1)$, und $\rho = \frac{r}{\beta}\sqrt{\gamma - 2\beta \sin(\varphi)}$.

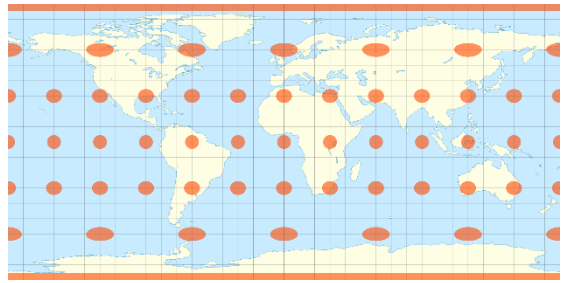


Kartenprojektion

Beispiele für längentreue Projektionen (hier nicht weiter im Detail besprochen):

- **Quadratischde Plattkarte**: längentreu entlang der Meridiane
  - Direkte Abbildung der geografischen Koordinaten in die Ebene
  - Grundidee: $x = \lambda$ und $y = \varphi$
  - Transformation:

$$x = (\lambda - \lambda_0) \cos \varphi_1$$
$$y = \varphi - \varphi_1$$



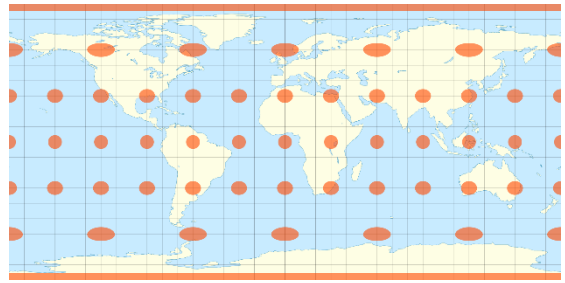
https://de.wikipedia.org/wiki/Plattkarte

Kartenprojektion

Beispiele für längentreue Projektionen (hier nicht weiter im Detail besprochen):

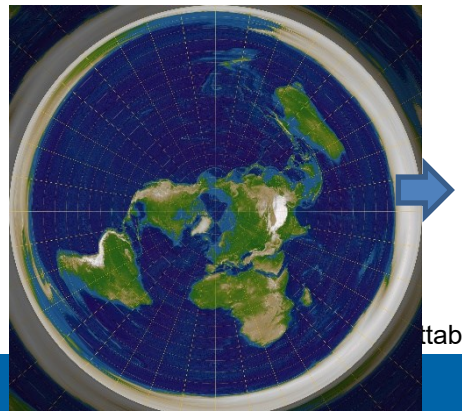
- **Quadratischde Plattkarte**: längentreu entlang der Meridiane
  - Direkte Abbildung der geografischen Koordinaten in die Ebene
  - Grundidee: $x = \lambda$ und $y = \varphi$
  - Transformation:

$$x = (\lambda - \lambda_0)\cos\varphi_1$$
$$y = \varphi - \varphi_1$$



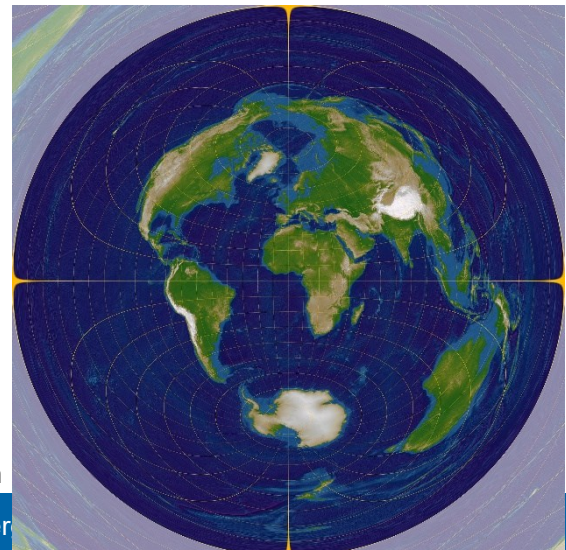
https://de.wikipedia.org/wiki/Plattkarte

- **Mittelabstandstreue Azimutalprojektion:** längentreu von zentralem Punkt aus





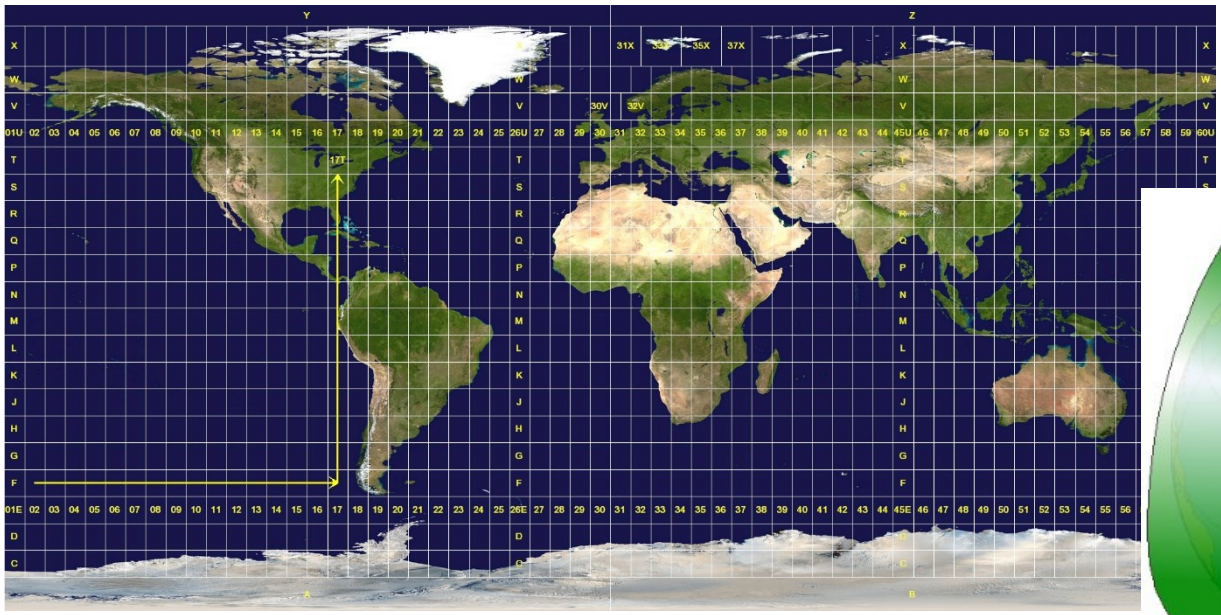
ttabstandstreue_Azimutalprojektion



Kartenprojektion

Sehr häufig verwendet: **UTM (Universal Transverse Mercator)**, Skalierungsfaktor α>0.9996

- Aufteilung der Erde in Zonen von ca. 6° Breite
- Jede Zone nutzt eine transversale Mercator Schittzylinderprojektion
- Referenzellipsoid ist GRS80 (nicht die Kugeloberfläche, wie bei den vorher angegebenen Tranformationsformeln)

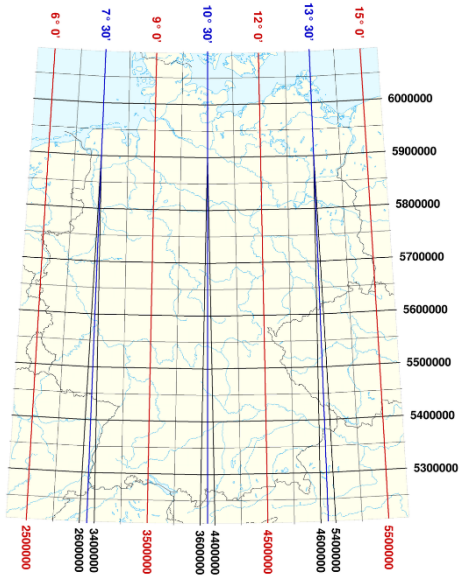




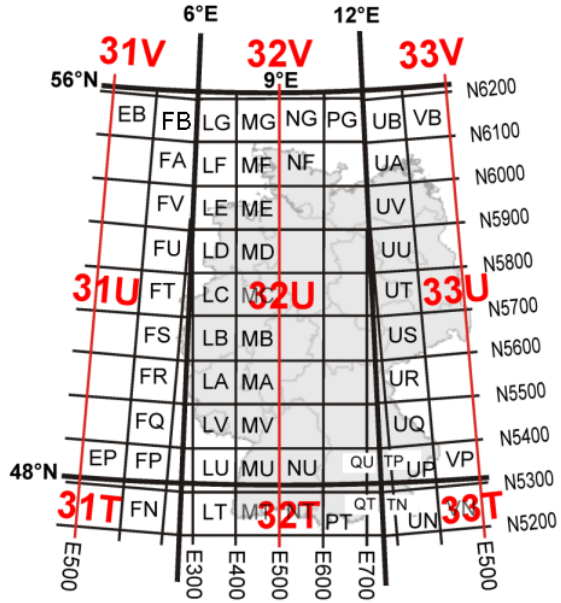
Kartenprojektion

Ebenfalls sehr häufig verwendet: **Gauss-Krüger**; nutzt ebenfalls transversale Mercator Projektion; Unterschiede zu UTM**:**

- Aufteilung der Erde in Zonen von ca. 3° Breite
- Referenzellipsoid ist Bessel



Gauss-Krüger



UTM

Koordinatenumwandlung

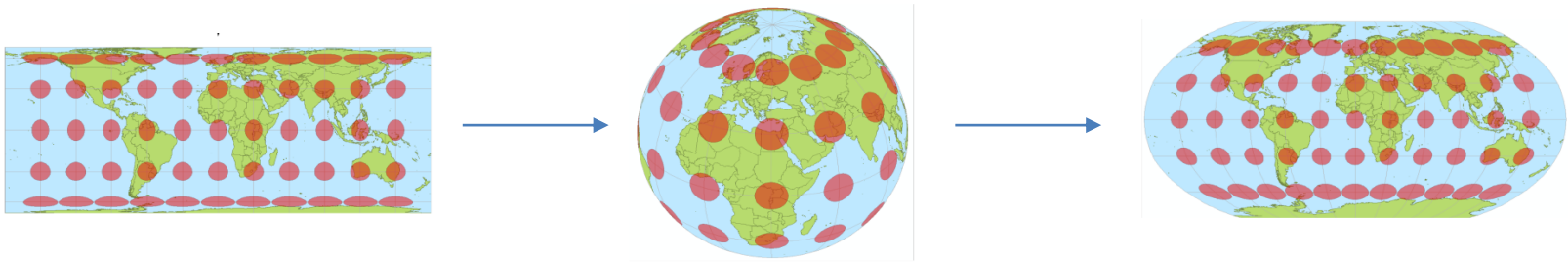
Georeferenzierung umfasst (grob gesagt) den Workflow, welcher den externen Raumbezug der eigentlichen Daten in den Raumbezug des GIS-Projektes umwandelt. Es muss der externe Raumbezug (z.B. geographische Koordinaten (λ,φ) oder Pixel in einem Luftbild) in ein üblicherweise durch (x,y)-Koordinaten gegebenen Raumbezug des GIS Projektes (lokales Koordinatensystem).

Wir unterscheiden oft zwei Koordinatenoperation:

- **Koordinatenumwandlung**
- Koordinatentransformartion

$$(x', y') \xrightarrow{T_1^{-1}} (\lambda, \varphi) \xrightarrow{T_2} (x'', y'')$$

Mit den bekannten Kartenprojektionen T_1 und T_2.



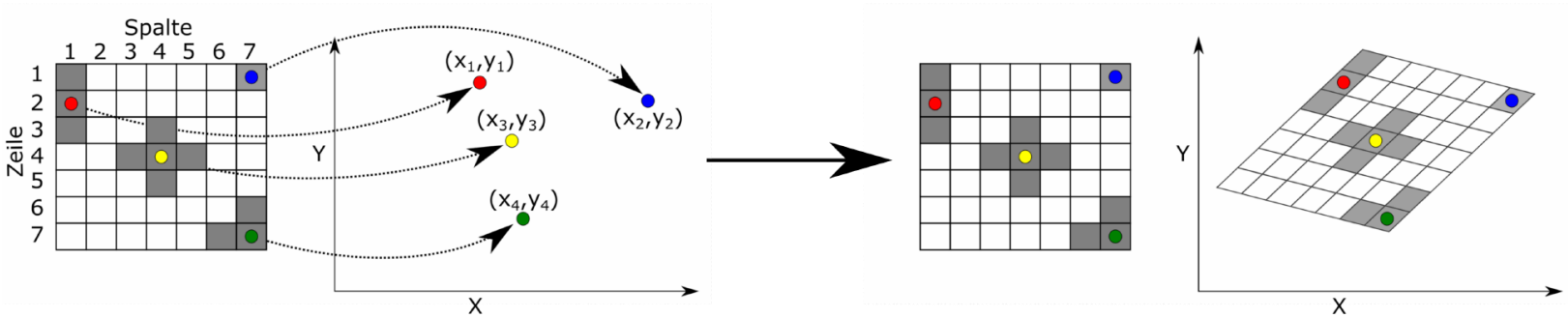
Koordinatentransformation

Georeferenzierung umfasst (grob gesagt) den Workflow, welcher den externen Raumbezug der eigentlichen Daten in den Raumbezug des GIS-Projektes umwandelt. Es muss der externe Raumbezug (z.B. geographische Koordinaten (λ,φ) oder Pixel in einem Luftbild) in ein üblicherweise durch (x,y)-Koordinaten gegebenen Raumbezug des GIS Projektes (lokales Koordinatensystem).

Wir unterscheiden oft zwei Koordinatenoperation:

- Koordinatenumwandlung
- **Koordinatentransformation**

Koordinatentransformationen werden verwendet, wenn kein expliziter Formalismus bekannt ist (z.B. zwei Rasterbilder liegen jeweils nur in relativen Pixelkoordinaten vor).



Transformation basiert auf **Passpunkten** in den jeweiligen lokalen Koordinaten

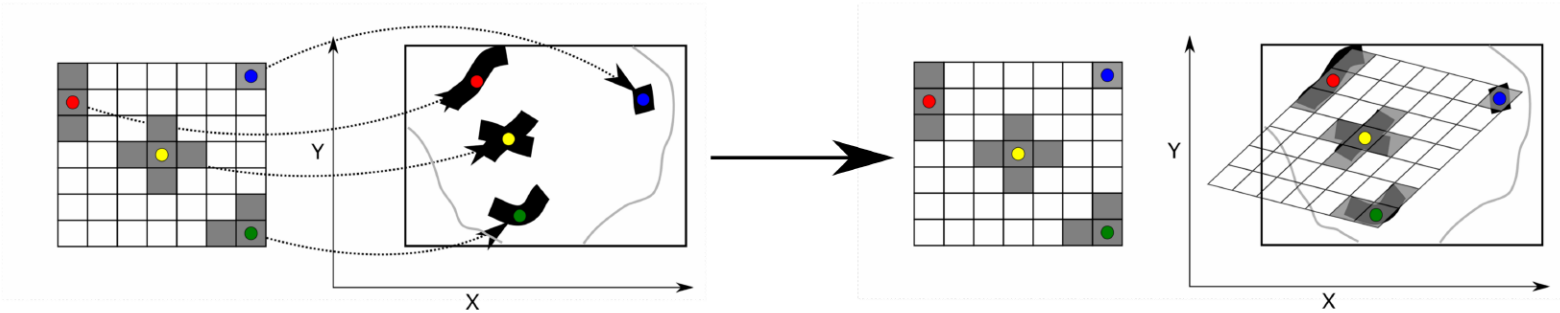
Koordinatentransformation

Georeferenzierung umfasst (grob gesagt) den Workflow, welcher den externen Raumbezug der eigentlichen Daten in den Raumbezug des GIS-Projektes umwandelt. Es muss der externe Raumbezug (z.B. geographische Koordinaten (λ,φ) oder Pixel in einem Luftbild) in ein üblicherweise durch (x,y)-Koordinaten gegebenen Raumbezug des GIS Projektes (lokales Koordinatensystem).

Wir unterscheiden oft zwei Koordinatenoperation:

- Koordinatenumwandlung
- **Koordinatentransformation**

Koordinatentransformationen werden verwendet, wenn kein expliziter Formalismus bekannt ist (z.B. zwei Rasterbilder liegen jeweils nur in relativen Pixelkoordinaten vor).



Transformation basiert auf Zuweisung von Objekten zu Passpunkten im Zielkoordinatensystem

Koordinatentransformation

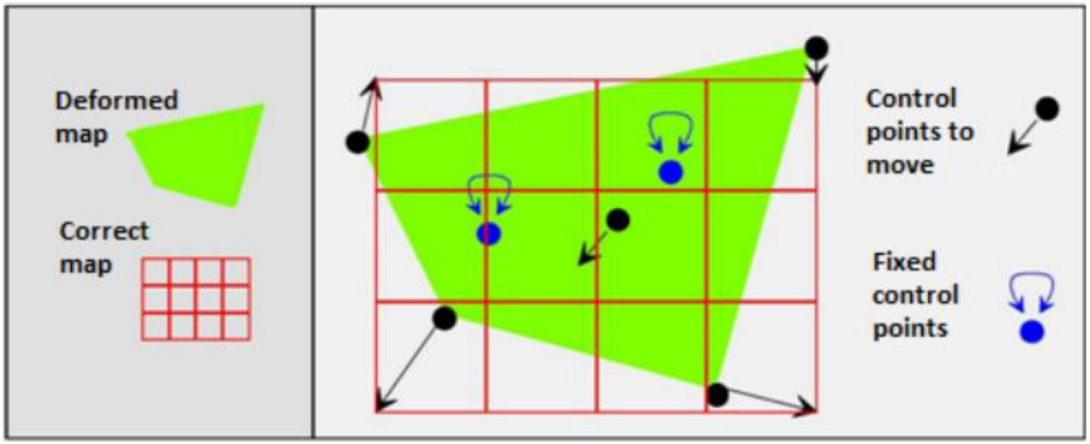
Affin-lineare Abbildungen können zwischen lokalen Koordinaten (x,y) und (x',y') in zwei verschiedenen Koordinatensystemen transformieren.

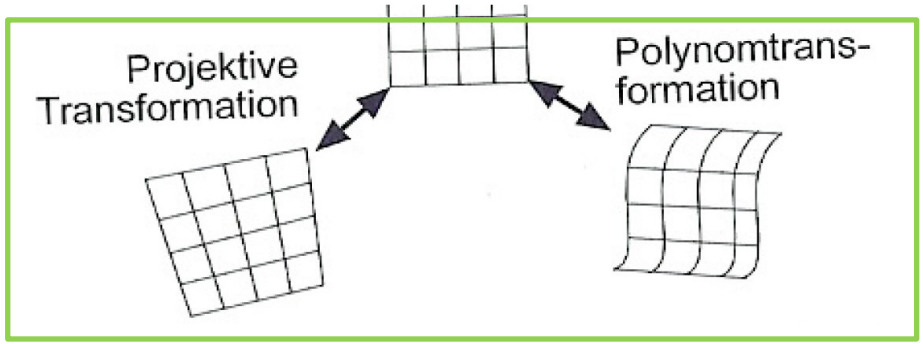
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{M}$$

wobei $\mathbf{M}$

$$\mathbf{M} = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix}$$

die z.B. au





Koordinatentransformation

Affin-lineare Abbildungen können zwischen lokalen Koordinaten (x,y) und (x',y') in zwei verschiedenen Koordinatensystemen transformieren.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{M} \begin{pmatrix} x \\ y \end{pmatrix} + \mathbf{v},$$

wobei $\mathbf{M}$ eine Matrix ist

$$\mathbf{M} = \begin{pmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{pmatrix} = \mathbf{RS}$$

die z.B. aus Rotationen $\mathbf{R}$ und Skalierungen und Scherungen $\mathbf{S}$ besteht.

Auch als Polynome 1. Grades ausdrückbar:

$$x' = m_{1,1}x + m_{1,2}y + v_1$$
$$y' = m_{2,1}x + m_{2,2}y + v_2$$

Koordinatentransformation

Verzerrungen und **nichtlineare Effekte** benötigen Transformationen höherer Ordnung (z.B. Polynome vom Grad zwei, drei, etc.): Für Grad zwei sähe es wie folgt aus

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{M} \begin{pmatrix} x \\ y \end{pmatrix} + (x, y)\, \mathbf{N} \begin{pmatrix} x \\ y \end{pmatrix} + \mathbf{v},$$

wobei $\mathbf{M} \in \mathbb{R}^{2\times 2}$ und $\mathbf{N} \in \mathbb{R}^{2\times 2\times 2}$ geeignete Matrizen sind.

vgl. Matlab Beispiel transformations.m / coordinateTransformations.m

Auch als Polynome 2. Grades ausdrückbar:

$$x' = m_{1,1}x + m_{1,2}y + v_1 + n_{1,1,1}x^2 + n_{1,2,1}y^2 + n_{1,1,2}xy + n_{1,2,2}yx$$
$$y' = m_{2,1}x + m_{2,2}y + v_2 + n_{2,1,1}x^2 + n_{2,2,1}y^2 + n_{2,1,2}xy + n_{2,2,2}yx$$

mit

$$n_{1,1,2}xy + n_{1,2,2}yx = n_{1,.,2}xy$$
$$n_{2,1,2}xy + n_{2,2,2}yx = n_{2,.,2}xy$$

Koordinatentransformation

Sind die Matrizen **M** und **N** bekannt, so kann einfach zwischen den Koordinaten gewechselt werden. Sind Sie jedoch nicht bekannt, so müssen die Einträge erstmal durch Passpunkte bestimmt werden. Dazu ist das Lösen von (linearen und ggfs. nichtlinearen) Gleichungssystemen nötig: Etwa bei drei Passpunkten im affin-linearen Fall

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} m_{1,1} \\ m_{1,2} \\ v_1 \end{pmatrix}, \qquad \begin{pmatrix} y'_1 \\ y'_2 \\ y'_3 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} m_{2,1} \\ m_{2,2} \\ v_2 \end{pmatrix}$$

wobei $m_{1,1}, \ldots, m_{2,2}$ die (unbekannten) Einträge der Matrix $\mathbf{M}$ sind und v_1, v_2 die (unbekannten) Einträge des Vektors $\mathbf{v}$ sind.

Koordinatentransformation

Sind die Matrizen **M** und **N** bekannt, so kann einfach zwischen den Koordinaten gewechselt werden. Sind Sie jedoch nicht bekannt, so müssen die Einträge erstmal durch Passpunkte bestimmt werden. Dazu ist das Lösen von (linearen und ggfs. nichtlinearen) Gleichungssystemen nötig: Etwa bei drei Passpunkten im affin-linearen Fall

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} m_{1,1} \\ m_{1,2} \\ v_1 \end{pmatrix}, \qquad \begin{pmatrix} y'_1 \\ y'_2 \\ y'_3 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} m_{2,1} \\ m_{2,2} \\ v_2 \end{pmatrix}$$

wobei $m_{1,1}, \ldots, m_{2,2}$ die (unbekannten) Einträge der Matrix $\mathbf{M}$ sind und v_1, v_2 die (unbekannten) Einträge des Vektors $\mathbf{v}$ sind.

Gibt es ungünstige Positionen der Passpunkte in obigem Fall? Falls ja, was wäre ein Beispiel und welche Probleme könnten auftauchen?

Koordinatentransformation

Sind die Matrizen **M** und **N** bekannt, so kann einfach zwischen den Koordinaten gewechselt werden. Sind Sie jedoch nicht bekannt, so müssen die Einträge erstmal durch Passpunkte bestimmt werden. Dazu ist das Lösen von (linearen und ggfs. nichtlinearen) Gleichungssystemen nötig: Etwa bei drei Passpunkten im affin-linearen Fall

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} m_{1,1} \\ m_{1,2} \\ v_1 \end{pmatrix}, \qquad \begin{pmatrix} y'_1 \\ y'_2 \\ y'_3 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} m_{2,1} \\ m_{2,2} \\ v_2 \end{pmatrix}$$

wobei $m_{1,1}, \ldots, m_{2,2}$ die (unbekannten) Einträge der Matrix $\mathbf{M}$ sind und v_1, v_2 die (unbekannten) Einträge des Vektors $\mathbf{v}$ sind.

Gibt es ungünstige Positionen der Passpunkte in obigem Fall? Falls ja, was wäre ein Beispiel und welche Probleme könnten auftauchen?

Wieviele Passpunkte braucht man jeweils für ein affin-lineare, eine quadratische (Grad zwei), und kubische (Grad drei) Transformation?

3, 6, bzw. 10 (da jeweils 6, 12, bzw. 20 Freiheitsgrade)

Koordinatentransformation

Sind die Matrizen **M** und **N** bekannt, so kann einfach zwischen den Koordinaten gewechselt werden. Sind Sie jedoch nicht bekannt, so müssen die Einträge erstmal durch Passpunkte bestimmt werden. Dazu ist das Lösen von (linearen und ggfs. nichtlinearen) Gleichungssystemen nötig: Etwa bei drei Passpunkten im affin-linearen Fall

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} m_{1,1} \\ m_{1,2} \\ v_1 \end{pmatrix}, \qquad \begin{pmatrix} y'_1 \\ y'_2 \\ y'_3 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} m_{2,1} \\ m_{2,2} \\ v_2 \end{pmatrix}$$

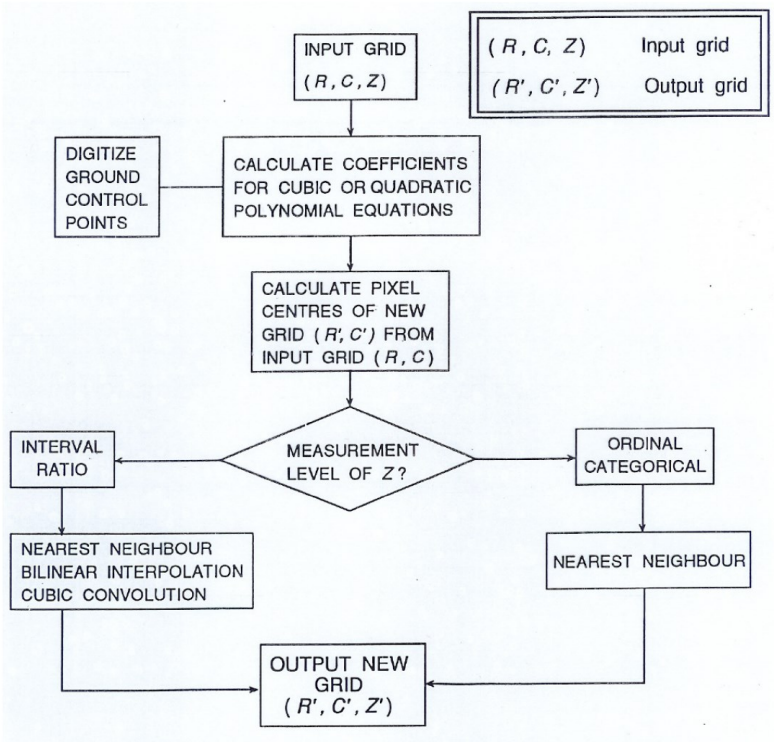
wobei $m_{1,1}, \ldots, m_{2,2}$ die (unbekannten) Einträge der Matrix $\mathbf{M}$ sind und v_1, v_2 die (unbekannten) Einträge des Vektors $\mathbf{v}$ sind.

Wie ändert sich die Anzahl der Freiheitsgrade z.B. im affin-linearen Fall, wenn wir die Dimension erhöhen, etwa Dimension drei, vier, fünf? Wie ändert sich die Anzahl der nötigen Passpunkte?

12, 20, bzw. 30 (jeweils 4, 5, bzw. 6 Passpunkte nötig)

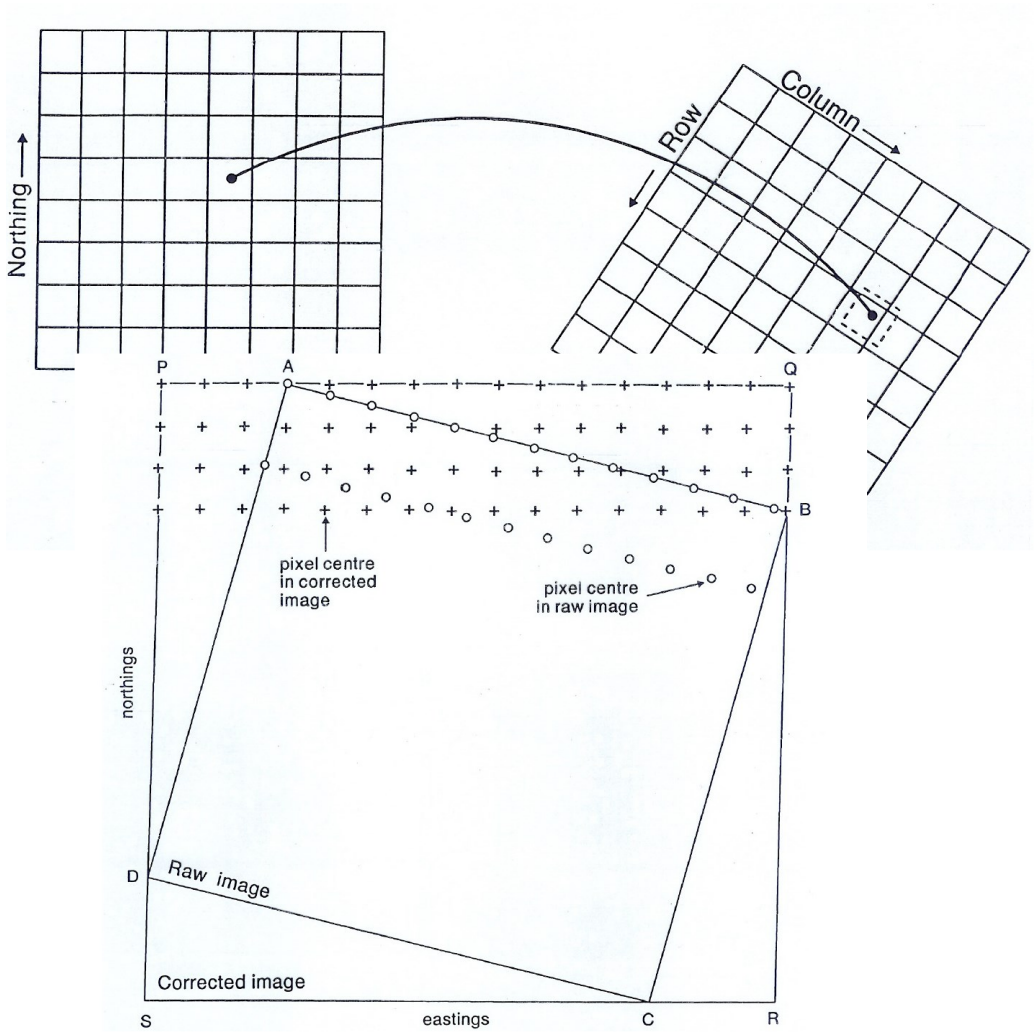
Georeferenzierung von Rasterdaten

Rasterdaten sind üblicherweise durch Attributwerte in Pixeln eines kartesischen Gitters. Nach Anwendung einer Koordinatentransformation befinden sich die transformierten Pixel in der Regel nicht mehr auf einem kartesischen Gitter und die Attributwerte müssen interpoliert werden.



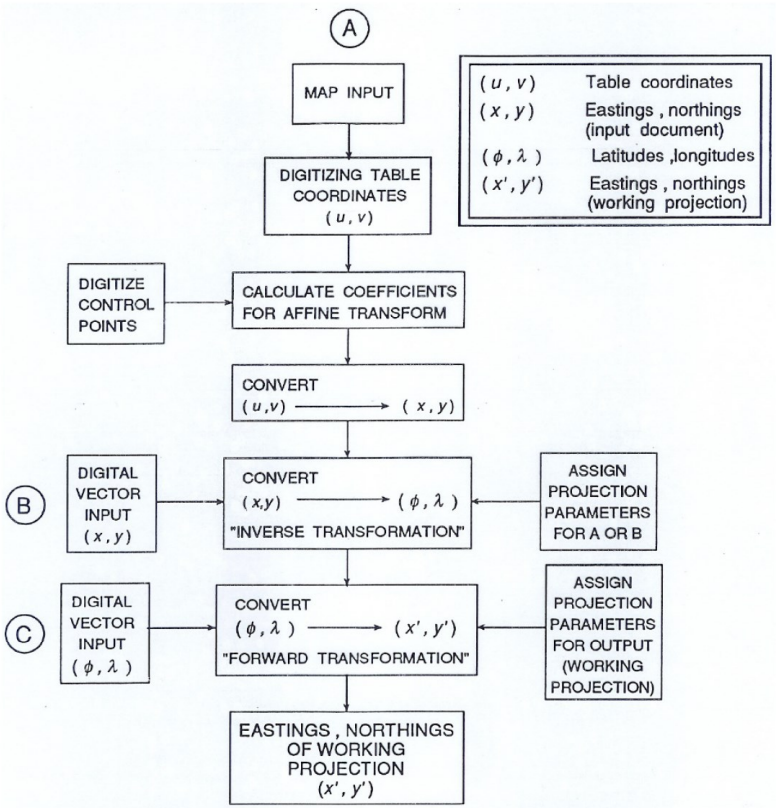
Bonham-Carter: Geographic Information Systems for Geoscientists: Modeling with GIS (1994)

Georeferenzierung von Rasterdaten



Georeferenzierung von Vektordaten

Vektordaten benötigen kein zugrundeliegendes kartesisches Gitter. Datensätze (hier z.B. A,B,C) können direct in ein gemeinsames Koordinatensystem überführt werden.



Bonham-Carter: Geographic Information Systems for Geoscientists: Modeling with GIS (1994)

Übersicht

1. Definitionen, Funktionen, Anwendungen (Vorlesung 1)
2. Koordinatensysteme und -transformationen (Vorlesung 2+3)
3. **Räumliche Datenmodellierung**
4. Vermaschungen
5. Räumliche Interpolation
6. Transformationen, Filtermethoden, Sonstiges

Räumliche Datenmodellierung

- **Datenmodell**: konzeptionelles Schema zu Organisation der Daten (z.B. Vektor- oder Rastermodell)
- **Datenstruktur**: Formen der Repräsentation eines Datenmodells (z.B. Matrixdarstellung für ein Rastermodell
- **Datenformat**: Möglichkeiten der Speicherung einer Datenstruktur (z.B. für eine Matrix im Rastermodell: array von durch Kommata getrennten Attributwerten)

Objekttypen

Räummliche Objekte können in verschiedener Gruppen sortiert oder anhand verschiedener Eigenschaften charakterisiert werden.

- **Dimension**: 0-d (Punktobjekte), 1-d (Linienobjekte), 2-D (Flächenobjekte), 3-D (Volumenobjekte), …; auch 2.5-D Objekte begrifflich möglich (2-D Objekte mit zusätzlichem räumlichen Parameter)

- **Natürliche vs. künstliche Objekte**:
  - natürliche Objekte = real auftretende Strukturen wie Flüsse, Gebäude, Gesteinskörper;
  - künstliche Objekte = von natürlichen physischen Objekten abgeleitete Darstellungen (Pixel, Isolinien, …) oder nicht-physische motivierte Objekte (Verwaltungsbezirke, Staatsgrenzen, …)

- **Diskrete vs. kontinuierliche Objekte**:
  - diskret = es können nur endlich viele Werte oder isolierte Werte angenommen werden (Lithologien, Einwohnerzahl, …),
  - kontinuierlich = alle ‘Zwischenwerte’ können ohne ‘Sprünge’ angenommen warden (Höhen, Stoffkonzentrationen, …)

    => vergl. Skalen-Begriff aus “Datenanalyse/Statistik”

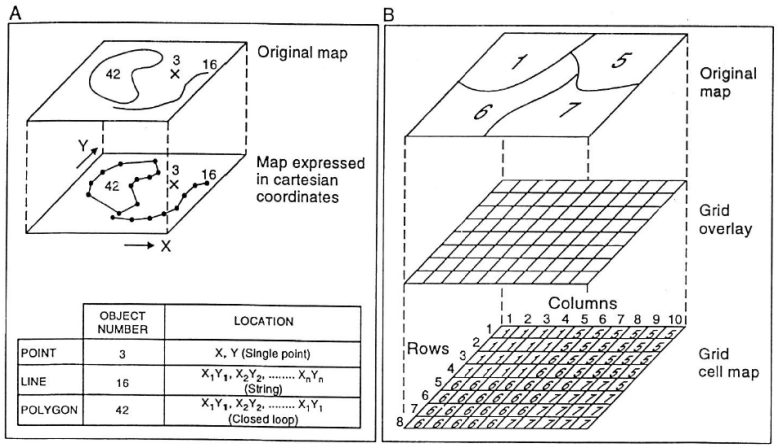
Objekttypen

- **Auflösungsbegrenzt vs. Definitionsbegrenzt**:
  - auflösungsbegrenzt = die Anzahl und Verteilung der Messdaten schränkt Aussagen über das Objekt ein (z.B. Genauigkeit der Küstenlinie hängt von Auflösung einer Luftbildaufnahme ab);
  - definitionsbegrenzt = zusätzliche Abhängigkeit von definierten Grenzwerten oder Attributen (z.B. Grenzwerte für Stoffkonmzentrationen oder Konturlinie basierend auf vorgegebener Höhe)

- **Unregelmässig vs. regelmässig**:
  - regelmässig = Menge gleichförmiger Objekte (Pixel, Daten auf regelmässigem Gitter, …);
  - unregelmäßig = Menge von Objekten unregelmässiger Form (Triangulierung mit nichtkongruenten Dreiecken, Waldgrenzen, …)

Raster- vs. Vektormodell

- **Rastermodell**:
  - gleichmässige Gitterstruktur
  - grundlegende Einheiten sind gleichförmig, üblicherweise Pixel (2-D) oder Voxel (3-D)
  - jeder grundlegenden Einheit wird ein Attributwert zugeordnet
  - grundlegenede Einheiten geben Auflösung vor



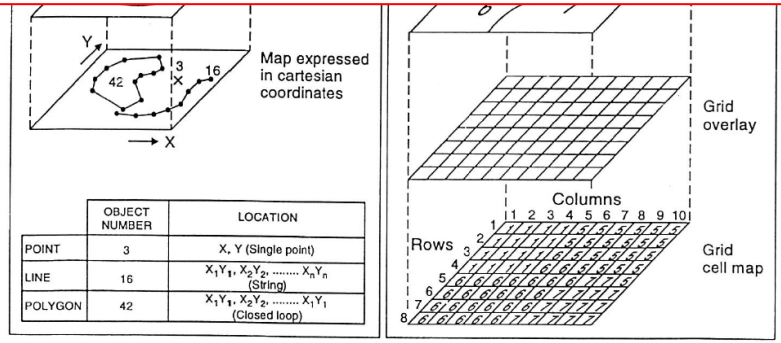
	OBJECT NUMBER	LOCATION
POINT	3	X, Y (Single point)
LINE	16	$X_1Y_1, X_2Y_2, \ldots\ldots X_nY_n$ (String)
POLYGON	42	$X_1Y_1, X_2Y_2, \ldots\ldots X_1Y_1$ (Closed loop)

Raster- vs. Vektormodell

- **Rastermodell**:
  - gleichmässige Gitterstruktur
  - grundlegende Einheiten sind gleichförmig, üblicherweise Pixel (2-D) oder Voxel (3-D)
  - jeder grundlegenden Einheit wird ein Attributwert zugeordnet
  - grundlegenede Einheiten geben Auflösung vor

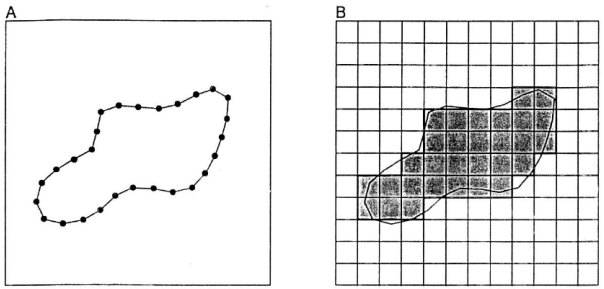
- **Vektormodell**:
  - Grundelegende Einheiten sind Vertexe (0-D Punktobjekte); und darauf aufbauend Linien/Polygonzüge, Polygone, …
  - Exakte Koordinaten für Vertexe, kein Auflösungsverlust
  - jedem Element kann ein oder mehrere Attributwerte zugeordnet werden

Sagen Sie Bitte niemals, einfach nur: „die Grundelemente eines Vektormodells sind Vektoren“!



Raster- vs. Vektormodell

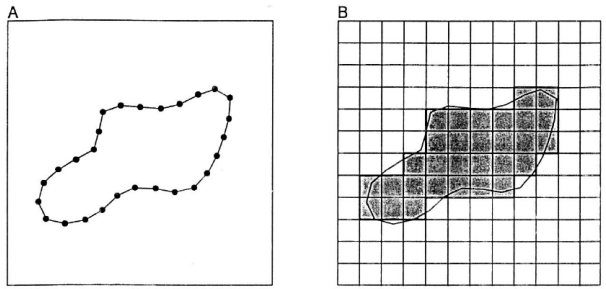
- **Rastermodell**:
  - ausschliesslich flächenhafte Betrachtung von Objekten: jedes Pixel beschreibt eine (möglicherweise sehr kleine) Fläche
  - größere Einzelobjekte lassen sich nur über Attributwerte voneinander abgrenzen
  - Datenerfassung einfach und effizient
  - Position und Nachbarschaftsbeziehungen sehr einfach über Gitteranordnung definiert
  - Unter Umständen sehr Speicheraufwendig



Raster- vs. Vektormodell

- **Rastermodell**:
  - ausschliesslich flächenhafte Betrachtung von Objekten: jedes Pixel beschreibt eine (möglicherweise sehr kleine) Fläche
  - größere Einzelobjekte lassen sich nur über Attributwerte voneinander abgrenzen
  - Datenerfassung einfach und effizient
  - Position und Nachbarschaftsbeziehungen sehr einfach über Gitteranordnung definiert
  - Unter Umständen sehr Speicheraufwendig

- **Vektormodell**:
  - Auflösungsunabhängig
  - größere Einzelobjekte als Einheit darstellbar
  - Effiziente Speicherung und Darstellung
  - Nachbarschaftsbeziehungen müssen festgelegt werden, aufwendigeres Anlegung der Datenstruktur



Raster- vs. Vektormodell

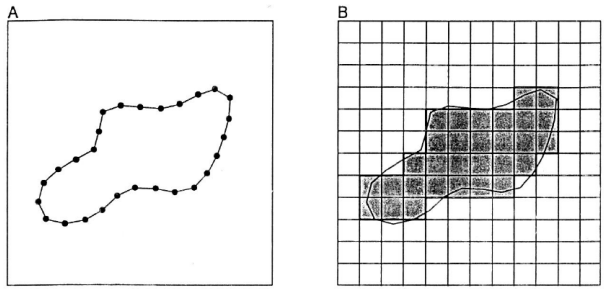
Bestimmung von Umfang und Fläche eines Objektes

- **Rastermodell**:
  - Fläche = Anzahl Pixel x Fläche pro Pixel
  - Umfang = Anzahl der Randpixel x Länge einer Pixelkante

  Ist die Formel für den Umfang so komplett korrekt? Bei welchen Pixeln in unterem Bild müsste aufgepasst werden?

  Obige Umfangs-Formel ist nur eine grobe Abschätzung, exakt wäre:

$$\text{Umfang} = \left(\sum_{\text{Anzahl Pixel}} \text{Anzahl der Nachbarpixel mit anderem Attribut} \right) \cdot \text{Kantenlänge}$$

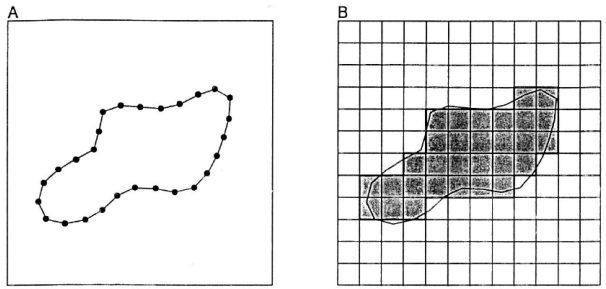


Raster- vs. Vektormodell

Bestimmung von Umfang und Fläche eines Objektes

- **Rastermodell**:
  - Fläche = Anzahl Pixel x Fläche pro Pixel
  - Umfang = Anzahl aller Nachbarpixel x Länge einer Pixelkante (Schätzung!)

- **Vektormodell**:
  - Fläche = Summe der Fläche der Teildreiecke
  - Umfang = Summe der Länge der Berandungslinien

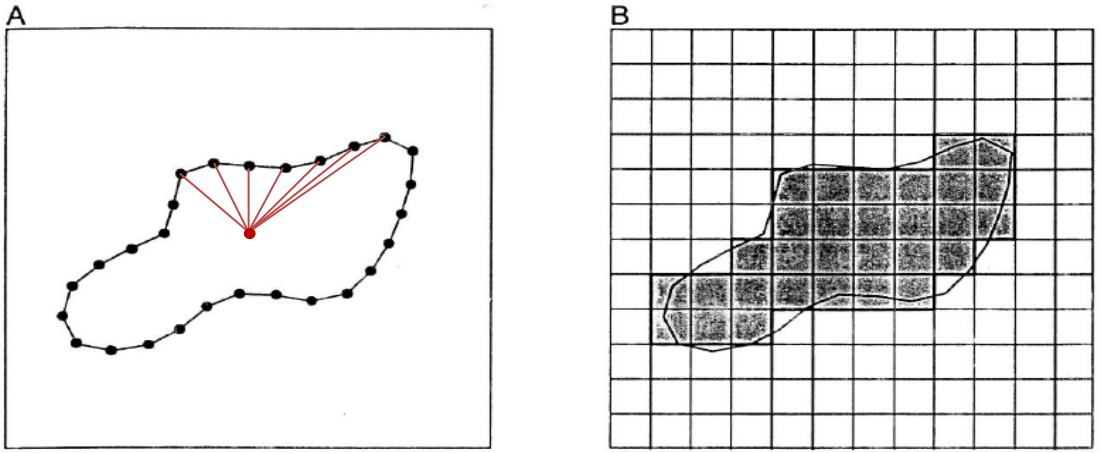


Raster- vs. Vektormodell

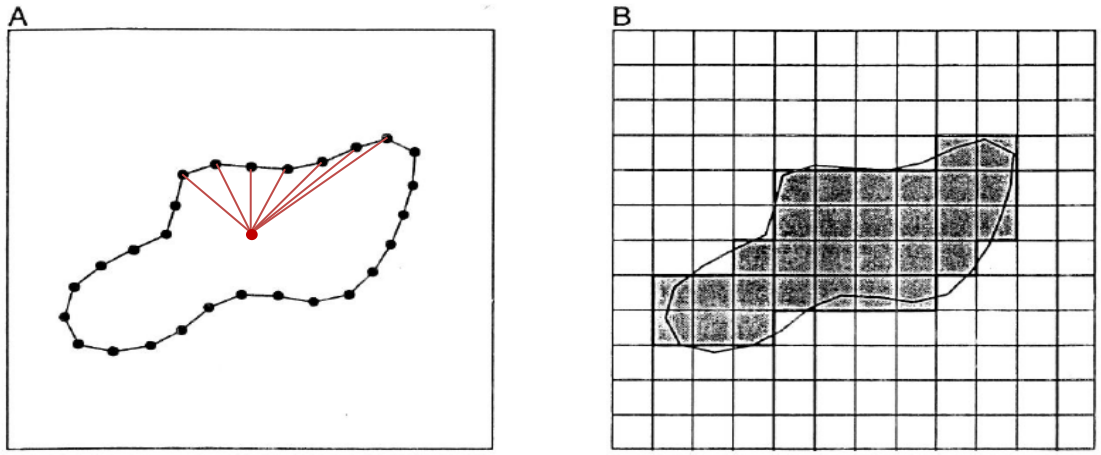
Bestimmung von Umfang und Fläche eines Objektes

- **Rastermodell**:
  - Fläche = Anzahl Pixel x Fläche pro Pixel
  - Umfang = Anzahl der Randpixel x Länge einer Pixelkante (Schätzung!)

- **Vektormodell**:
  - Fläche = Summe der Fläche der Teildreiecke
  - Umfang = Summe der Länge der Berandungslinien



Raster- vs. Vektormodell



Die Fläche eines Dreiecks mit Eckpunkten $(x_1, y_1), (x_2, y_2), (x, y)$ ist

$$\frac{1}{2}|(x - x_1)(y_1 - y_2) + (x_1 - x_2)(y_1 - y)| = \left|\det \begin{pmatrix} x & x_1 & x_2 \\ y & y_1 & y_2 \\ 1 & 1 & 1 \end{pmatrix}\right|.$$

Für das gesamte Polygon im Vektormodell ergibt sich als Fläche

$\frac{1}{2} \sum_{i=1}^{n} |x_i y_{i+1} - x_{i+1} y_i|$.

Beachte, dass für letzteres Aufhebungseffekte bei der Summierung über alle Dreiecke eine Rolle spielen, so dass der Referenzpunkt (x, y) herausfällt.

Raster- vs. Vektormodell

Geoobjekte können über verschiedene Attribute beschrieben werden:

- räumliche Attribute
- zeitliche Attribute
- thematische Attribute

Die Attribute können in Attributtabellen zu den jeweiligen Objekten hinterlegt werden. Möglichkeit der Verknüpfung von Raster und Vektordaten.

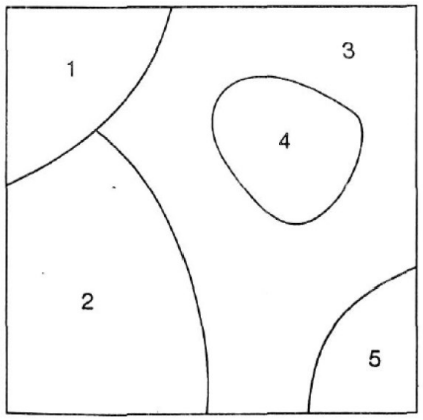
Raster- vs. Vektormodell

Geoobjekte können über verschiedene Attribute beschrieben werden:

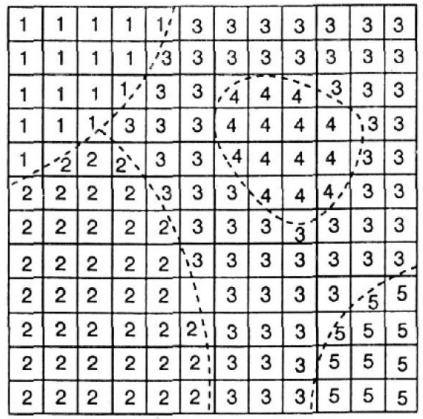
- räumliche Attribute
- zeitliche Attribute
- thematische A

Die Attribute k hinterlegt werden. Möglichkeit der \

A



B



C

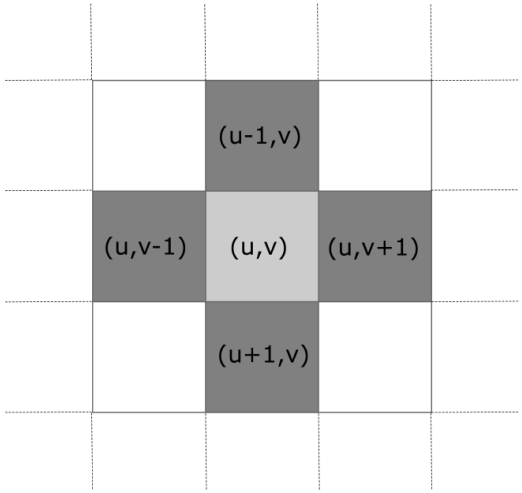
Polygon	Class	Rock type	Age	Name
1	15	sandstone	Late Pennsylvanian	Andrews Formation
2	6	limestone	Early Silurian	Barry Formation
3	3	shale	Middle Silurian	Clinton shale
4	14	granite	Devonian	Delta granite
5	14	granite	Devonian	Delta granite

Bonham-Carter: Geographic Information Systems for Geoscientists: Modeling with GIS (1994)

Rastermodell

Rasterdaten können sehr einfach als Matrix $A \in \mathbb{R}^{n \times m}$ hinterlegt werden. Die Einträge werden über Zeilen- und Spaltennummer angesprochen. Durch eine vorherige Georeferenzierung können die Pixelpositionen dann einfach in geographische Koordinaten umgerechnet werden.

Nachbarschaftsbeziehungen sind einfach über die benachbarten Pixel zu definieren.



Rastermodell

Rasterdaten können sehr einfach als Matrix $A \in \mathbb{R}^{n \times m}$ hinterlegt werden. Die Einträge werden über Zeilen- und Spaltennummer angesprochen. Durch eine vorherige Georeferenzierung können die Pixelpositionen dann einfach in geographische Koordinaten umgerechnet warden.

Nachbarschaftsbeziehungen sind einfach über die benachbarten Pixel zu definieren.

Wie werden die einzelnen Pixel (und damit auch Nachbarschaftsbeziehungen) angesteuert? Eine entsprechende **Datenstruktur** ist notwendig.

Kann einfach zwischen verschiedenen Layern/Bändern gewechselt werden?

Spalte (*column*)

Zeile (*row*)	1	2	3	4
1	a(1,1)	a(1,2)	a(1,3)	a(1,4)
2	a(2,1)	a(2,2)	a(2,3)	a(2,4)
3	a(3,1)	a(3,2)	a(3,3)	a(3,4)
4	a(4,1)	a(4,2)	a(4,3)	a(4,4)

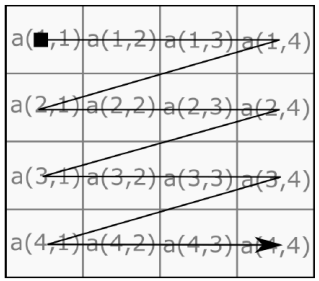
Band A

b(1,1)	b(1,2)	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)
b(4,1)	b(4,2)	b(4,3)	b(4,4)

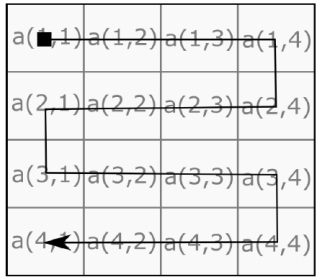
Band B

Rastermodell

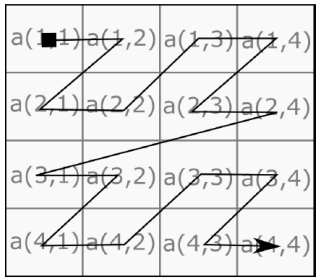
Oft ist es effizient und einfach, Matrixeinträge sequentiell in einer Zeile abzulegen. Dafür muss jedoch die Durchlaufvorschrift der Matrix bekannt sein!



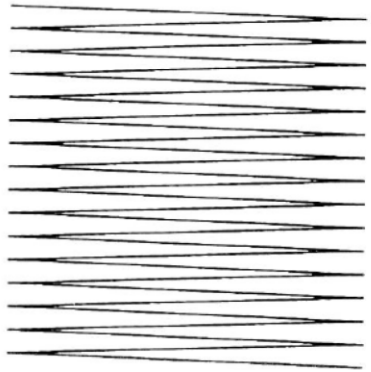
Band A

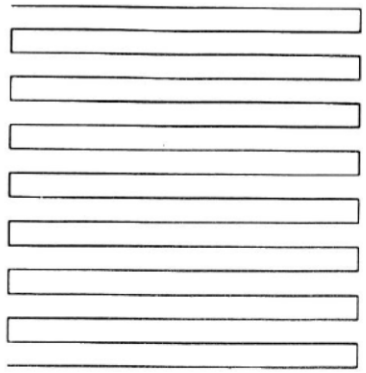


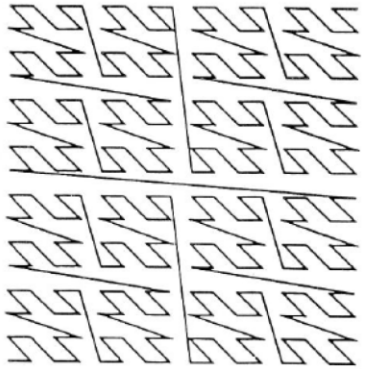
Band A

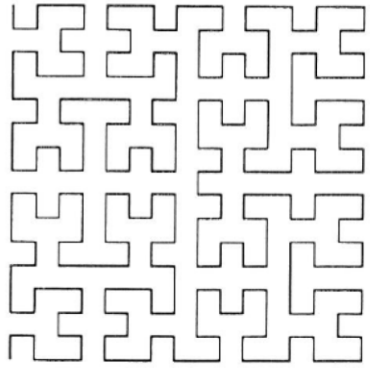


Band A


(a) Row Order

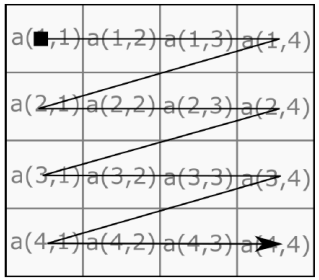

(b) Row-prime Order


(c) Morton Order

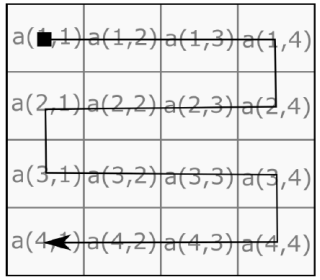

(d) Pi-Order

Rastermodell

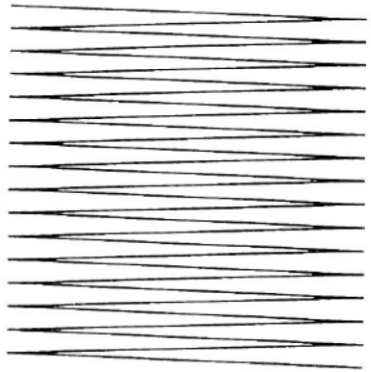
Oft ist es effizient und einfach, Matrixeinträge sequentiell in einer Zeile abzulegen. Dafür muss jedoch die Durchlaufvorschrift der Matrix bekannt sein!

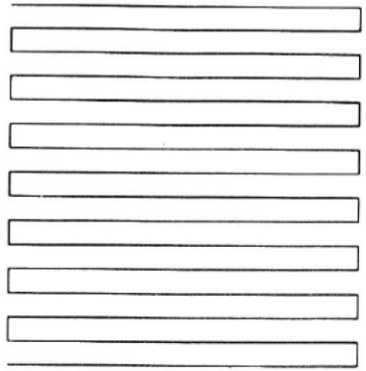


Band A



Band A

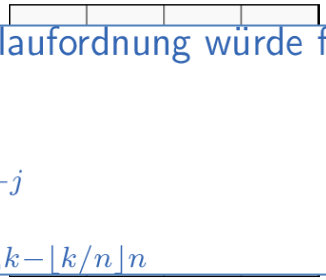

(a) Row Order

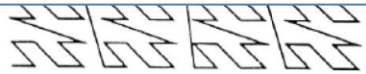

(b) Row-prime Order

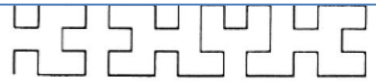
Welcher Durchlaufordnung würde folgende Zuweisung von Zeileneinträgen s_k zu Matrixeinträgen $a_{i,j}$ entsprechen?

$a_{i,j} \mapsto s_{(i-1)n+j}$

$s_k \mapsto a_{\lfloor k/n \rfloor + 1, k - \lfloor k/n \rfloor n}$


Band A


(c) Morton Order

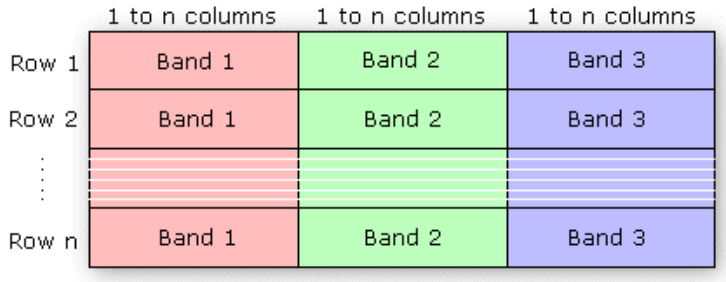

(d) Pi-Order

Rastermodell

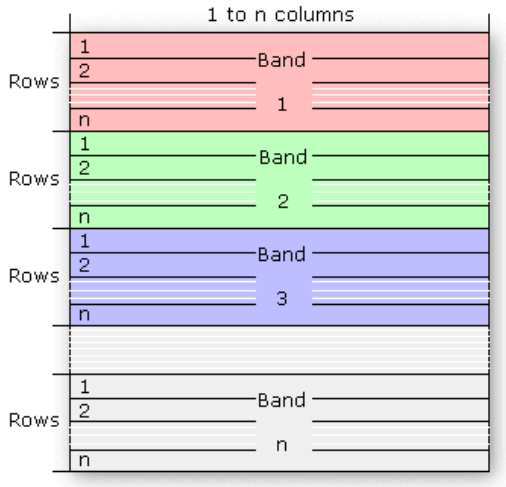
Angenommen wir wollen mehrere Bänder/Layer durchlaufen (z.B. bei hyperspektralen Daten). Welche Durchlaufvorschriften gibt es dann?

Nach dieser Kombination der Bänder/Layer kann wieder eines der vorherigen Durchlaufmuster verwendet werden.

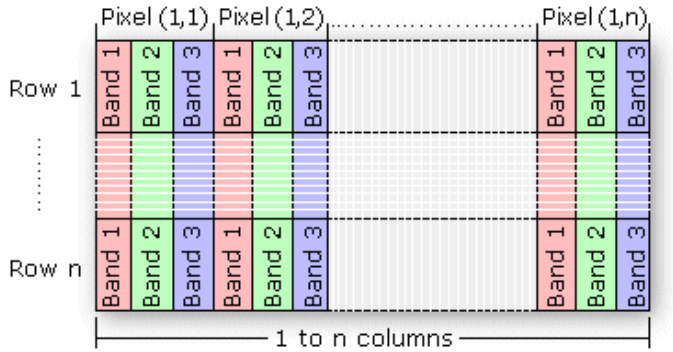
Band Interleaved by Line (BIL)



Band sequential (BSP)



Band Interleaved by Pixel (BIP)



https://desktop.arcgis.com/de/arcmap/10.3/manage-data/raster-and-images/bil-bip-and-bsq-raster-files.htm

Rastermodell

Große Datenmengen (z.B. hyperspektrale Daten) benötigen effektive speichereffiziente Methoden. Fortgeschrittene Beispiele sind z.B. JPEG (basierend auf wavelets) oder NASA fast lossless algortihm für hyperspektral Daten (www.techbriefs.com/component/content/article/tb/pub/techbriefs/information-sciences/49).

Hier wollen wir zwei einfache Methoden besprechen:

- **Lauflängenkodierung**

Statt Attributwerte für jedes Pixel zu speichern, werden die aufeinander Pixel mit gleichem Attributwert kombiniert und wie folgt gespeichert (abhängig von der Durchlaufvorschrift):

		1	2	3	4	5	6	7	8	9	10	Run-length encoding
Rows	1	A	A	A	A	B	B	B	A	A	A	(4,A),(3,B),(3,A)
	2	A	A	A	B	B	B	A	A	A	C	(3,A),(3,B),(3,A),(1,C)
	3	A	A	B	B	B	A	A	A	C	C	(2,A),(3,B),(3,A),(2,C)
	4	A	B	B	B	A	A	C	C	C	C	(1,A),(3,B),(2,A),(4,C)
	5	A	A	A	A	A	A	C	C	C	C	(6,A),(4,C)

Wie verhält sich der Speicheraufwand von Lauflänge zu Attributwert? Geht damit Informationsverlust einher? Kann man die Homogenitätsannahme aufweichen? Mit welchen Konsequenzen?

Rastermodell

- **Blockkodierung**

Statt einer zeilenweisen, könnte man auch auf flächige homogene Objekte abzielen. Zum Beispiel in dem man wie folgt codiert:

		1	2	3	4	5	6	7	8	9	10	Run-length encoding
Rows	1	A	A	A	A	B	B	B	A	A	A	(4,A),(3,B),(3,A)
	2	A	A	A	B	B	B	A	A	A	C	(3,A) … 1,C)
	3	A	A	B	B	B	A	A	A	C	C	(2,A),(3 …),(2,C)
	4	A	B	B	B	A	A	C	C	C	C	(1,A) … (2, … 4,C)
	5	A	A	A	A	A	A	C	C	C	C	(6,A),(4,C)

(1,1,2,A), (1,3,1,A), (1,4,1,A), (1,5,2,B), (1,7,1,B), (1,8,2,A), (1,10,1,A), (2,3,1,A), (2,4,2,B), …

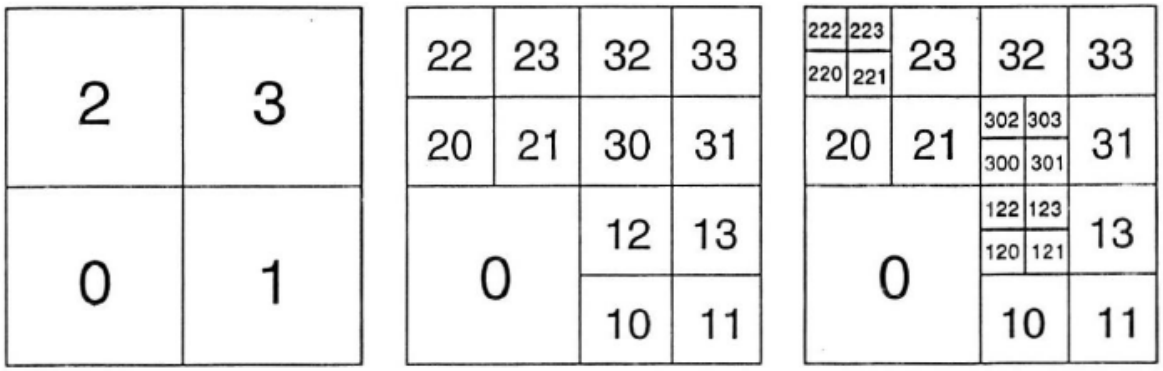
Die ersten zwei Einträge beschreiben die Zeile und Spalte der linken oberen Ecke eines Quadrates, der dritte Eintrag die Seitenlänge des Quadrates, der vierte Eintrag das Attribut.

Effizient, wenn viele großflächige homogene Gebiete vorliegen. Bei sehr komplexen Bildstruktur ist die Kompressionsrate gering.

Rastermodell

- **Quadtrees (Octrees in 3-D)**

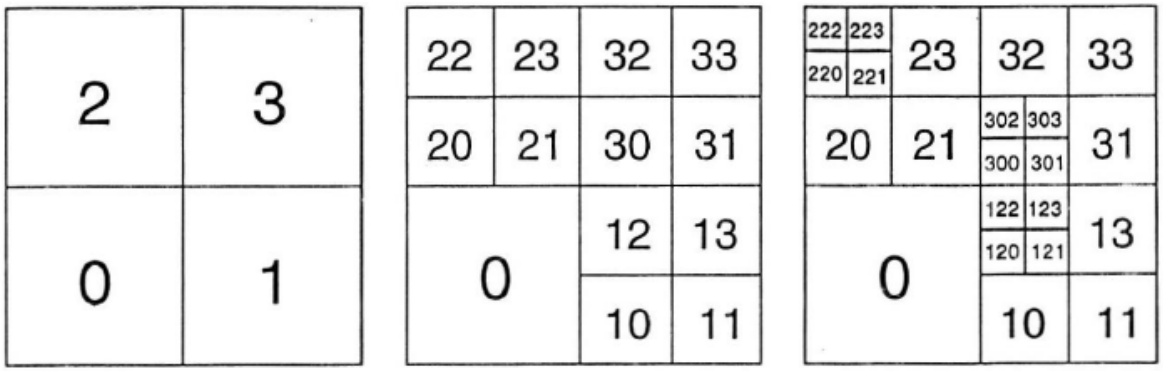
Ein weiteres auf flächige homogene Objekte abzielendes System. Sukzessive, hierarchische Unterteilung des Rasters in kleiner werdende Quadranten.



Rastermodell

- **Quadtrees (Octrees in 3-D)**

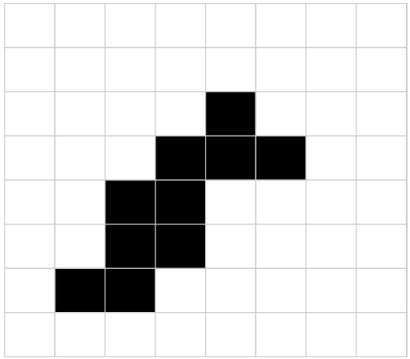
Ein weiteres auf flächige homogene Objekte abzielendes System. Sukzessive, hierarchische Unterteilung des Rasters in kleiner werdende Quadranten.

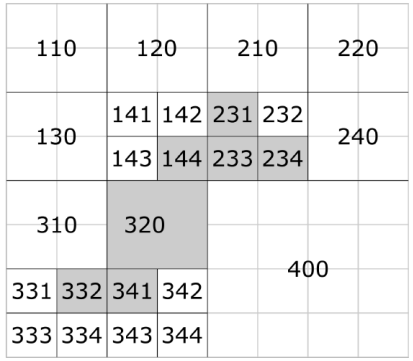


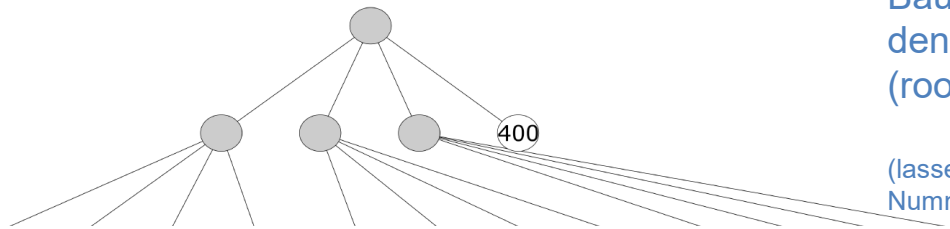
Es bietet sich an die Quadranten mit einem Zahlsystem zur Basis vier zu versehen, für eine intuitive Nummerierung der Quadranten.

Rastermodell

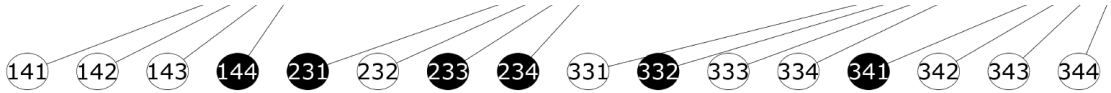
- **Quadtrees (Octrees in 3-D)**







https://www.maptiler.com/google-maps-coordinates-tile-bounds-projection/



Die Zerlegung des linken Rasters mittels Quadtrees sähe wie folgt aus:

(400,W), (110,W), (120,W), (130,W), (320,B),…, (344,W)

Die weißen Flächen müssten bei einem binären Bild theoretisch nicht mit gespeichert werden. Die Endknoten des hierarchischen Baums nennt man auch Blattknoten, den obersten Knoten Wurzelknoten (root node).

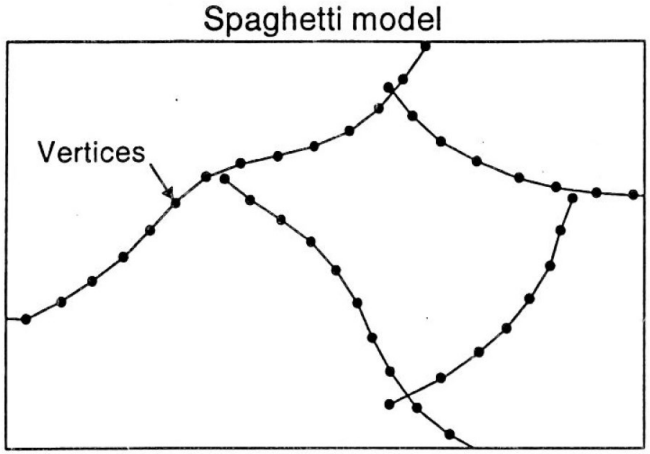
(lassen Sie sich nicht durch die geänderte Nummerierung der Quadranten verwirren)

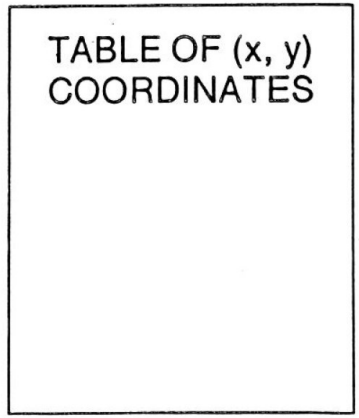
Wie verhält sich Speicheraufwand von Quadtrees zur vorher beschriebenen Blockkodierung?

Vektormodell

Grundlegende Strukturen sind Vertexe (0-D). Darauf aufbauend gibt es Linienzüge/Polygonzüge (1-D), Polygone (2-D), etc.

Die einfachste Datenstruktur ist die **Sphagettistruktur**:

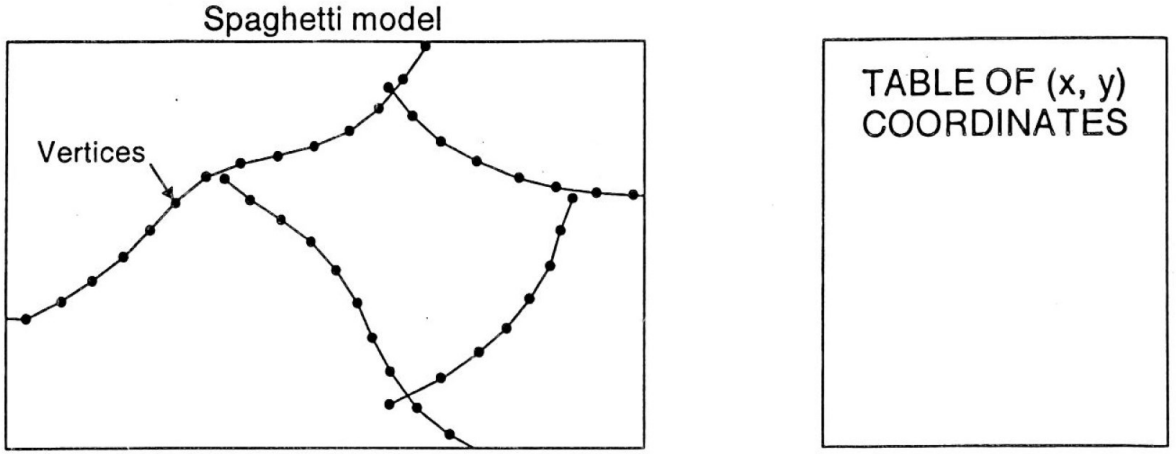




Vektormodell

Grundlegende Strukturen sind Vertexe (0-D). Darauf aufbauend gibt es Linienzüge/Polygonzüge (1-D), Polygone (2-D), etc.

Die einfachste Datenstruktur ist die **Sphagettistruktur**:



Einzelne Objekte werden als einfache Listen von Koodinaten der beinhalteten Vertices hinterlegt (Reihenfolge spielt eine Rolle). Bei Polygonen muss der letzte Vertex mit dem ersten übereinstimmen.

Vektormodell

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}

Einzelne Objekte werden als blosse Listen von Koodinaten der beinhalteten Vertexe hinterlegt (Reihenfolge spielt eine Rolle). Bei Polygonen muss der letzte Vertex mit dem ersten übereinstimmen.

Es können zusätzlich Attribute hinterlegt werden. Die Attribute können in separeter Liste hinterlegt werden.

Vektormodell

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.

Einzelne Objekte werden als bloss beinhalteten Vertexe hinterlegt (Reihenfolge spielt eine Rolle). tzte Vertex mit dem ersten übereinstimmen.

Es können zusätzlich Attribute hinterlegt werden. Die Attribute können in separeter Liste hinterlegt werden.

Vektormodell

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.

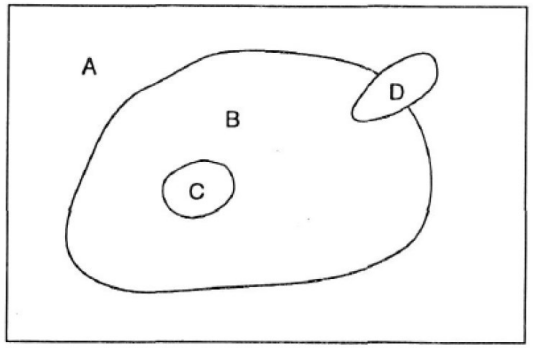
Einzelne Objekte werden als bloss [illegible] (Reihenfolge spielt eine Rolle). [illegible] übereinstimmen.

Es können zusätzlich Attribute hinterlegt werden. Die Attribute können in separeter Liste hinterlegt werden.

Vektormodell

Objekte in der Sphagettistruktur sind unabhängig voneinander.

- Dadurch entstehen Redundanzen (Vertexe, Linien, etc. werden eventuell mehrfach gespeichert).
- Wichtige Vorteile des Vektormodells z.B. bei Abfragen kommen nicht zum Tragen (es müssen schlussendlich wieder geometrische Information verarbeitet werden)
- Polygonen muss Priorität zugewiesen werden, um z.B. bei Plotten zu verhindern, dass grosse Polygone kleinere Polygone (etwa Inseln/Löcher) überdecken



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

Vorteil: sehr effiziente Darstellung auf grafischen Systemen, da alle Daten direkt dargestellt werden können. Es ist kein zusätzlicher Suchaufwand in mehreren Tabellen notwendig.

Vektormodell

Topologische Datenstrukturen: Darstellung von “Nachbarschafts”-Beziehungen zwischen Objekten (geometrische Eigenschaften und Distanzen spielen keine grosse Rolle, sondern nur die relative Lage und Beziehung zueinander)

- *Grund-Elementtypen*: Punkte/Vertex, Kante (verbindet zwei Punkte), Fläche (begrenzt durch Kanten), Volumen (begrenzt durch Flächen). Alle Objekte von Dimension d>0 sind begrenzt durch Elemente von Dimension d-1

Vektormodell

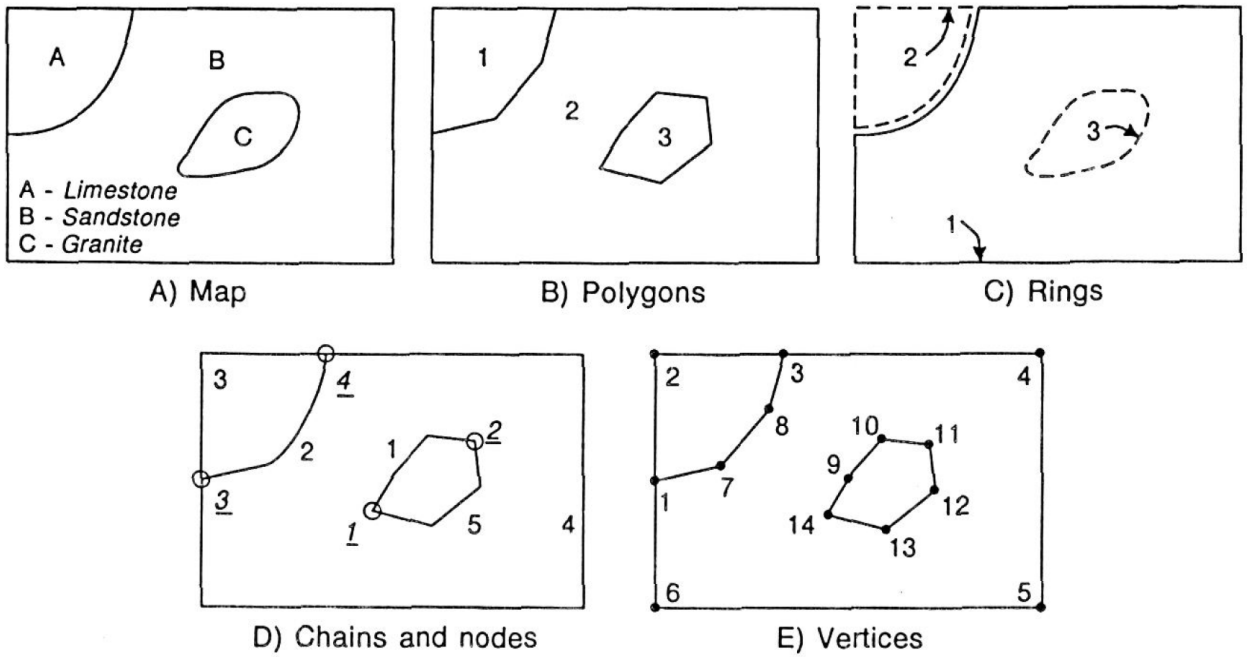
Topologische Datenstrukturen: Darstellung von “Nachbarschafts”beziehungen zwischen Objekten (geometrische Eigenschaften und Distanzen spielen keine grosse Rolle, sondern nur die relative Lage und Beziehung zueinander)

- *Grund-Elementtypen*: Punkte/Vertex, Kante (verbindet zwei Punkte), Fläche (begrenzt durch Kanten), Volumen (begrenzt durch Flächen). Alle Objekte von Dimension d>0 sind begrenzt durch Elemnte von Dimension d-1
- Grund-Elementtypen erlauben Defintion weiterer *topologischer Elemente*:
  - **Linie/Linienzug/Bogen (line/arc)**: geordnete, zusammenhängende Menge von Kanten mit einem End- und einem Anfangsknoten
  - **Knoten (Node)**: Punkt an dem eine Linie beginnt/endet oder an dem sich zwei oder mehrere Linien treffen
  - **Kette/Bogen (chain/arc)**: Linien, die Teil der Genze eines Flächenobjektes sind
  - **Ring**: Grenze eines Flächenenlements, welche aus mehreren Ketten/Bögen besteht
  - **Polygon**: Flächenelement begrenzt durch ein oder mehrere Ringe (mindestens ein äußerer Ring)
    - Einfaches Polygon: nur ein äußerer Ring
    - Komplexes Polygon: ein oder mehrere innere Ringe, zusätzlich zu einem äußeren Ring

Vektormodell

Topologische Datenstrukturen: Darstellung von “Nachbarschafts”beziehungen zwischen Objekten (geometrische Eigenschaften und Distanzen spielen keine grosse Rolle, sondern nur die relative Lage und Beziehung zueinander)

- *Grund-Elementtype*
  Kanten), Volumen
  durch Elemnte von
- Grund-Elementtype
  - **Linie/Linienz**
    Kanten mit ei
  - **Knoten (Noc**
    mehrere Linie
  - **Kette/Bogen**
  - **Ring**: Grenze
  - **Polygon**: Flä
    äußerer Ring
    - Einfach
    - Komple
      äußere



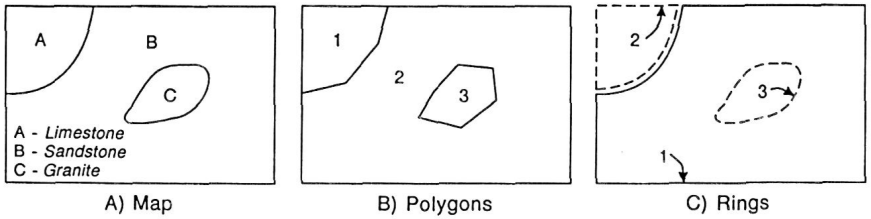
Vektormodell

Darstellung der topologischen Struktur mittels Listen/Tabellen (z.B. van-Roessel-Topologie):

- Polygon Topologie Tabelle
- Ring Topologie Tabelle
- Ketten Topologie Tabelle
- Knoten-zu-Punkt Tabelle
- Kette-zu-Punkt Tabelle
- Koordinatentabelle

Vektormodell

Darstellung der topologischen Struktur mittels Listen/Tabellen (z.B. van-Roessel-Topologie):



C. Chain topology table

Chain #	Start Node	Stop Node
1	1	2
2	3	4
3	4	3
4	4	3
5	1	2

F. Coordinates of vertices table (part only)

Vertex #	X	Y
1	x_1	y_1
2	x_2	y_2
.	.	.
.	.	.
14	x_{14}	y_{14}

D. Node-to-vertex table

Node #	Vertex #
1	14
2	11
3	1
4	3

E. Chain-to-vertex table

Chain #	Vertex #	Vertex Sequence #
1	14	1
1	9	2
1	10	3
1	11	4
2	1	1
2	7	2
2	8	3
2	3	4
3	3	1
3	2	2
3	1	3
4	3	1
4	4	2
4	5	3
4	6	4
4	1	5
5	14	1
5	13	2
5	12	3
5	11	4

A. Polygon topology table

Polygon #	Ring #	Ring Sequence #
1	2	1
2	1	1
2	3	2
3	3	1

B. Ring topology table

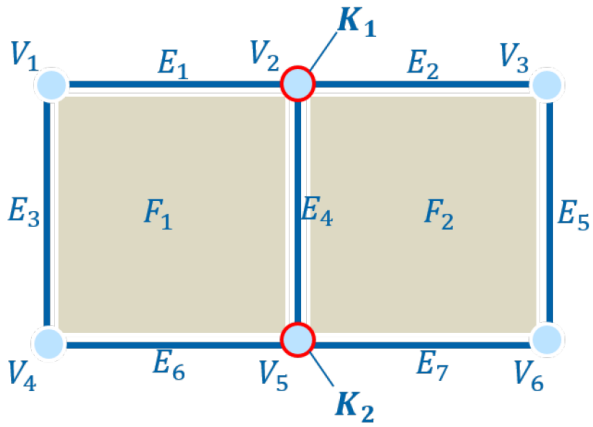
Ring #	Chain #	Chain Sequence #
2	3	1
2	2	2
1	2	1
1	4	2
3	1	1
3	5	2

Vektormodell

Für welche (beispielhaften) Abfragen könnte solch ein topologische Struktur sinnvoll/hilfreich sein?

- Finde alle Grenzschichten zwischen Granit und Kalkstein
- Entferne alle Kanten zwischen Polygonen mit gleichen Attributeigenschaften
- Finde alle Punkte, an denen sich mehr als zwei Kontinentalplatten treffen

Vektormodell



Linien/Ketten:

$L_1 = \{E_1, E_3, E_6\}$
$L_2 = \{E_4\}$
$L_3 = \{E_2, E_5, E_5\}$

Ringe:

$R_1 = \{L_1, L_2\}$
$R_2 = \{L_2, L_3\}$

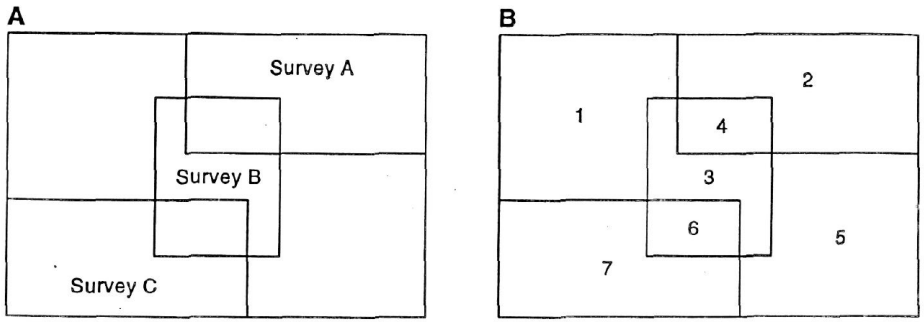
Inzidenz- und Adjazenzmetrizen als Möglichkeit zur mathematischen Beschreibung der Nachbarschaftsbeziehungen

$$I_{VE} = \begin{pmatrix} 1&0&1&0&0&0&0 \\ 1&1&0&1&0&0&0 \\ 0&1&0&0&1&0&0 \\ 0&0&1&0&0&1&0 \\ 0&0&0&1&0&1&1 \\ 0&0&0&0&1&0&1 \end{pmatrix} \qquad A_V = \begin{pmatrix} 1&1&0&1&0&0 \\ 1&0&1&0&1&0 \\ 0&1&0&0&0&1 \\ 1&0&0&0&1&0 \\ 0&1&0&1&0&1 \\ 0&0&1&0&1&0 \end{pmatrix}$$

Vektormodell

Ist jede Realisierung eines Vektormodells für solche Abfragen sinnvoll?

Es sollte nach Möglichkeit keine sich überlappenden Polygone geben. Erreichbar z.B. durch **Planar Enforcement** (oder allgemeiner, planare Vermaschungen)



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

Bonham-Carter: Geographic Information Systems for Geoscientists: Modeling with GIS (1994)

Raster- vs. Vektormodell

Abschliessender Vergleich:

- Rastermodell sehr gut geeignet zur Überlagerung von Layern und zur Bildverarbeitung (aber oft speicherintensiv)
- Vektormodell mit Sphagettistruktur: gut geeignet und effizient zur Visualisierung von Grenzen, z.B. in der Kartographie
- Vektormodell mit topologischer Struktur: sehr gut geeignet für räumliche Abfragen und Verbindungsinformationen (aber oft aufwendig anzulegen)

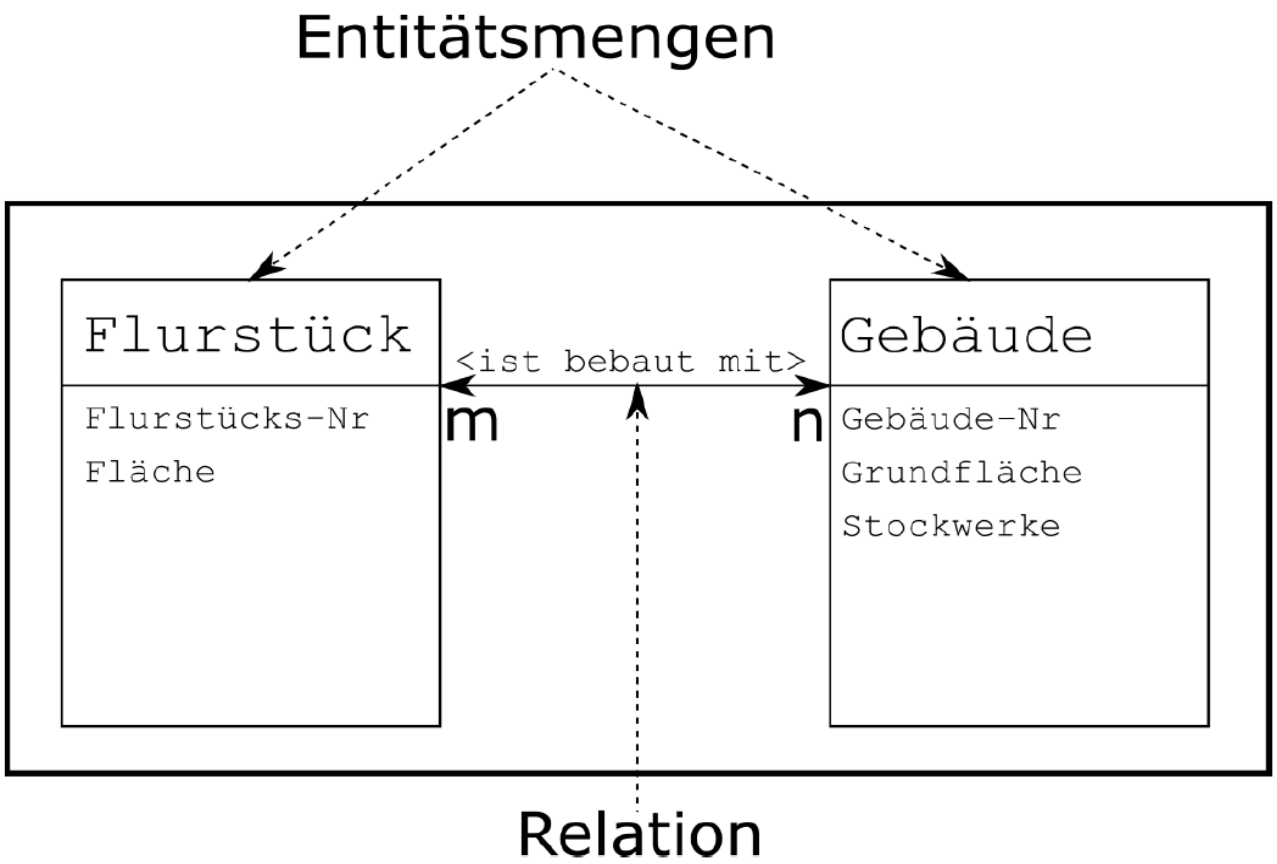
Weitere Datenmodelle

Entitäten-Relationen-Modell:

- Entität ist ein abstraktes Objekt, welches durch einen Satz von Attributen bestimmt wird
- Entitäten gleichen Typs weisen den gleichen Satz von Attributen auf
- Entitäten können durch Relationen verbunden werden:
  - 1-1 Relationen: eine Entität von Typ A ist eineindeutig mit einer Entität vom Typ B verbunden
  - 1-n Relation: eine Entität von Typ A ist mit n Entitäten vom Typ B verbunden
  - m-n Relation: m Entitäten von Typ A sind mit n Entitäten vom Typ B verbunden

Weitere Datenmodelle

Entitäten-Relationen-Modell:



Weitere Datenmodelle

Entitäten-Relationen-Modell:

- Entität ist ein abstraktes Objekt, welches durch einen Satz von Attributen bestimmt wird
- Entitäten gleichen Typs weisen den gleichen Satz von Attributen auf
- Entitäten können durch Relationen verbunden warden:
  - 1-1 Relationen: eine Entität von Typ A ist eineindeutig mit einer Entität vom Typ B verbunden
  - 1-n Relation: eine Entität von Typ A ist mit n Entitäten vom Typ B verbunden
  - m-n Relation: m Entitäten von Typ A sind mit n Entitäten vom Typ B verbunden

Graphen-basierte Modelle:

- Beziehungen zwischen Entitäten werden in gerichteter Form dargestellt
- hierarchische und vernetzte Darstellung möglich

Weitere Datenmodelle

Entitäten-Relationen-Modell:

- Entität ist ein abstraktes Objekt, welches durch einen Satz von Attributen bestimmt wird
- Entitäten Gleichen Typs weisen den Gleichen Satz von Attributen auf
- Entitäten können durch Relationen verbunden warden:
  - 1-1 Relationen: eine Entität von Typ A ist eineindeutig mit einer Entität vom Typ B verbunden
  - 1-n Relation: eine Entität von Typ A ist mit n Entitäten vom Typ B verbunden
  - m-n Relation: m Entitäten von Typ A sind mit n Entitäten vom Typ B verbunden

Graphen-basierte Modelle:

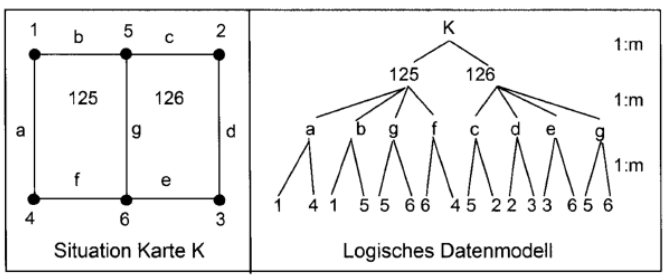


Abbildung 6.43: Hierarchisches Datenmodell.

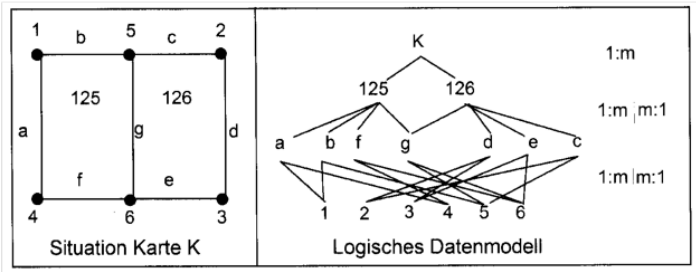
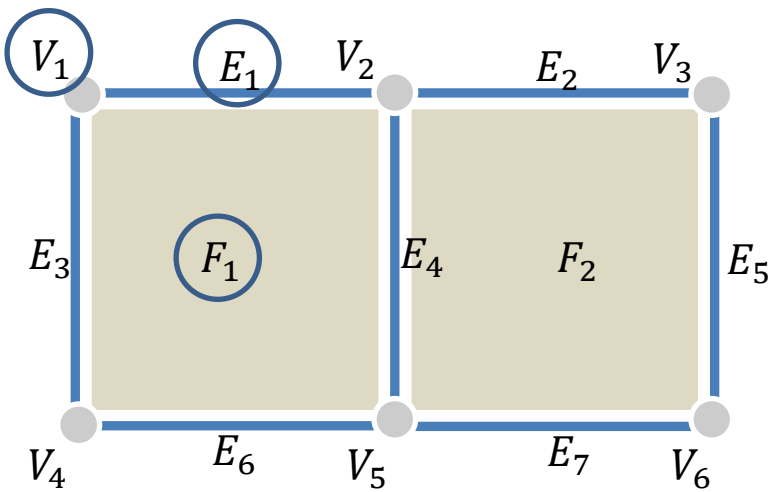
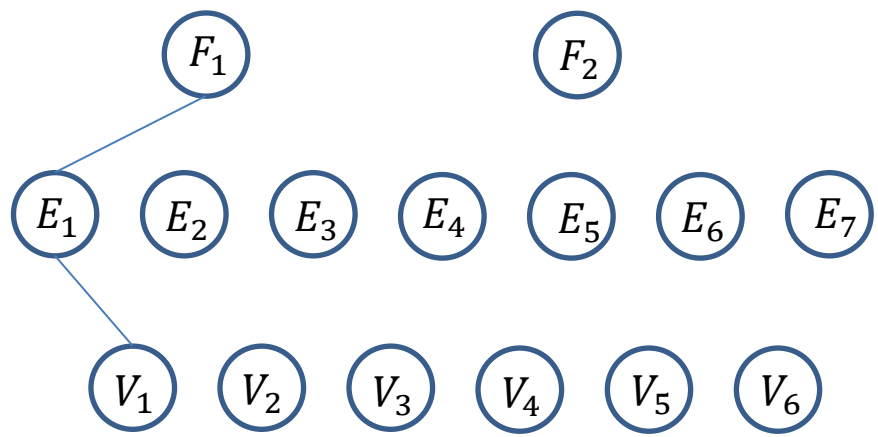


Abbildung 6.44: Netzwerkartiges Datenmodell.

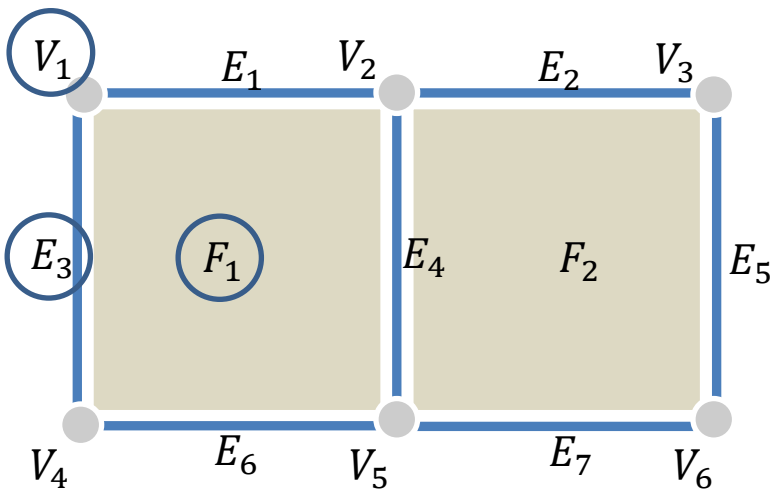
Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell



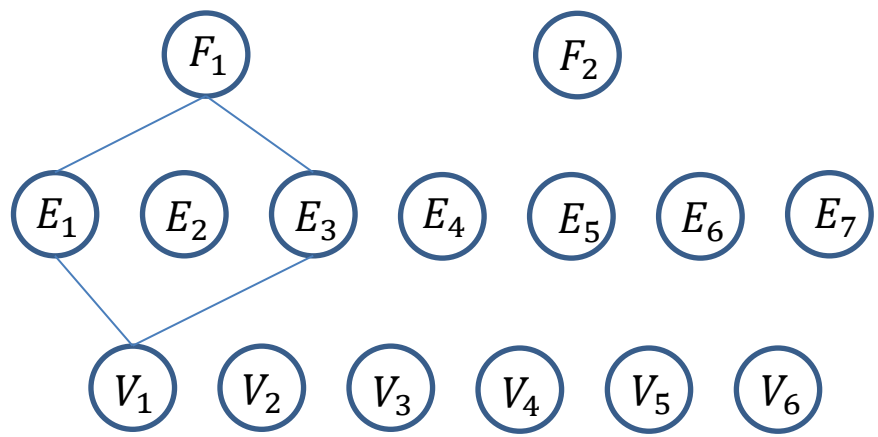
$\{V_1, E_1, F_1\}$



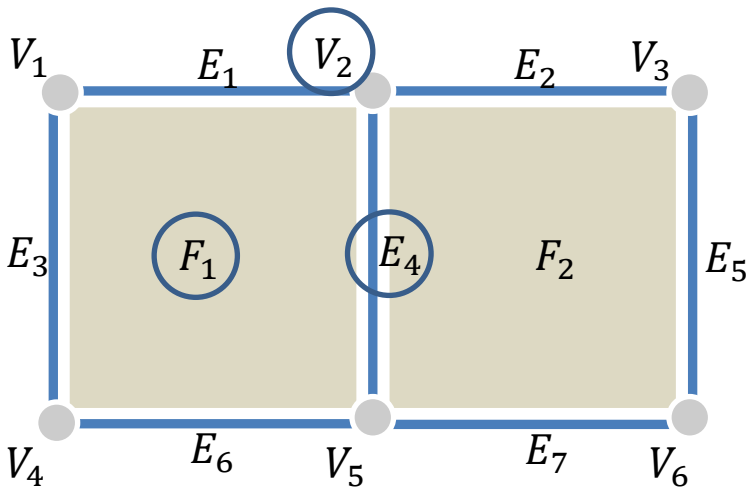
Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell



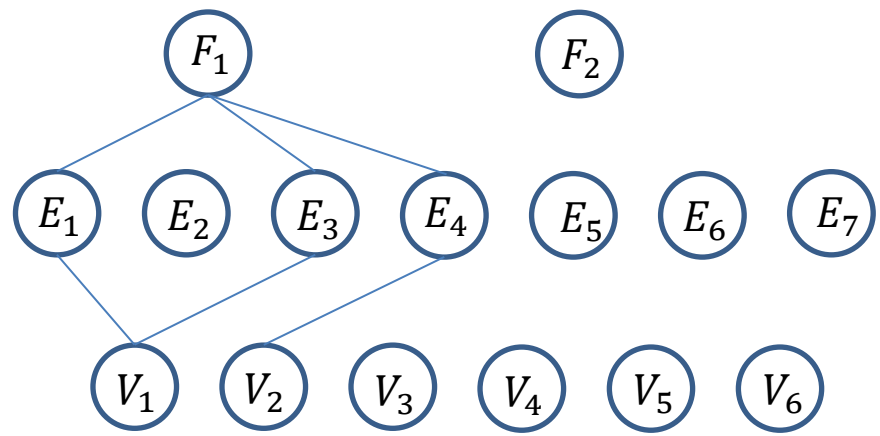
$\{V_1, E_3, F_1\}$



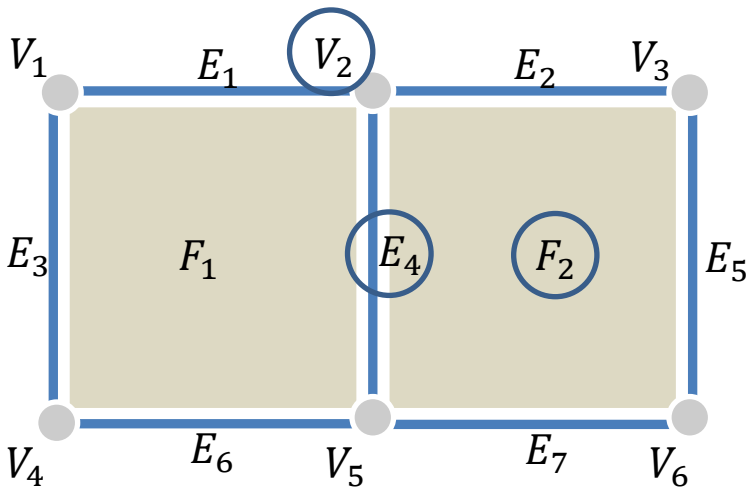
Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell



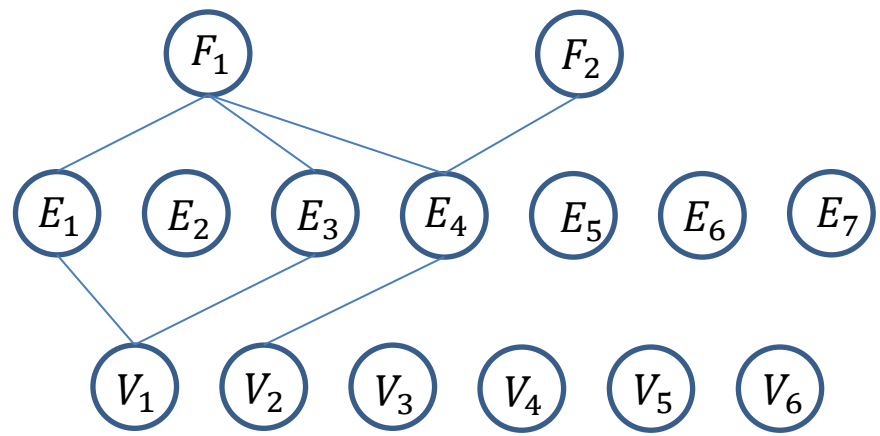
$\{V_2, E_4, F_1\}$



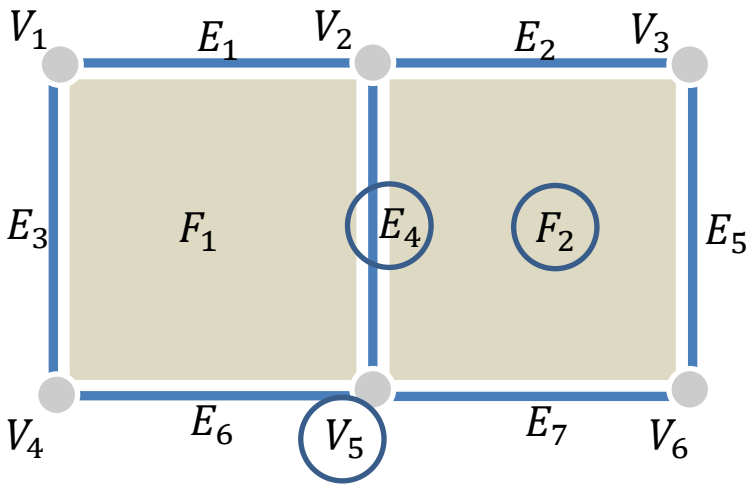
Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell



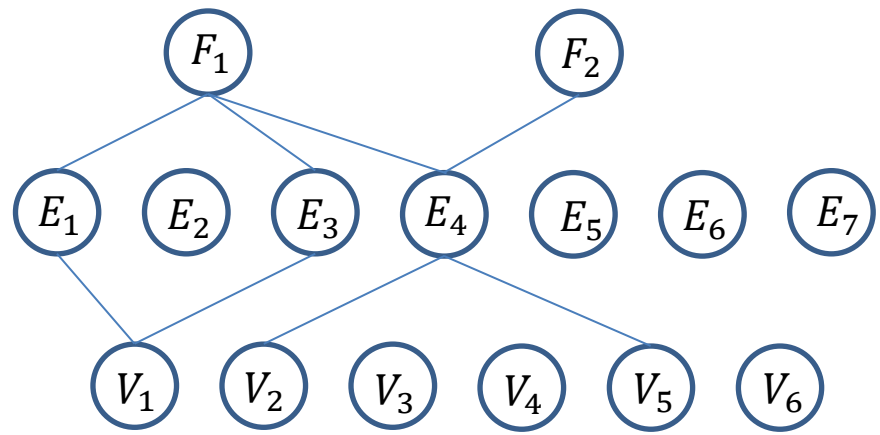
$\{V_2, E_4, F_2\}$



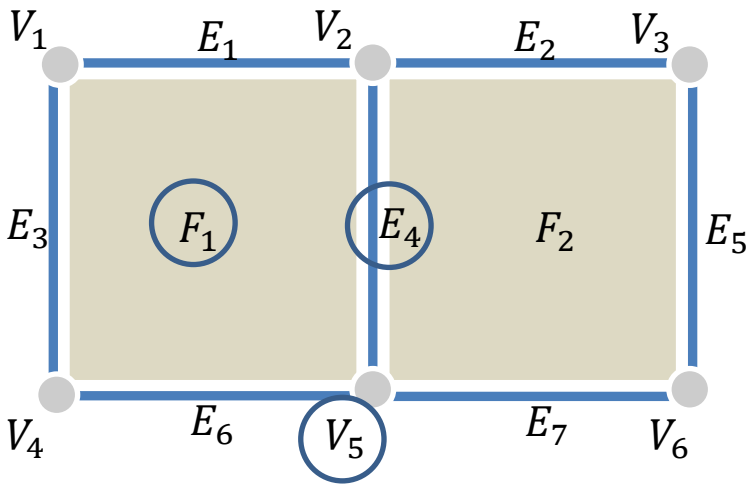
Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell



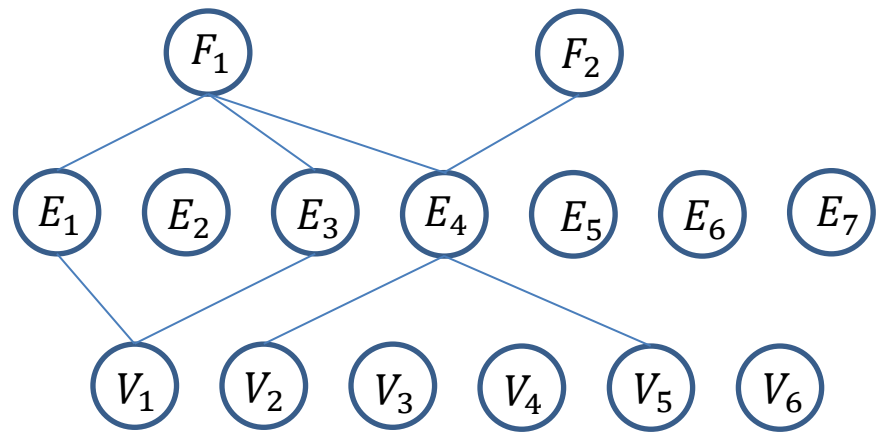
$\{V_5, E_4, F_2\}$



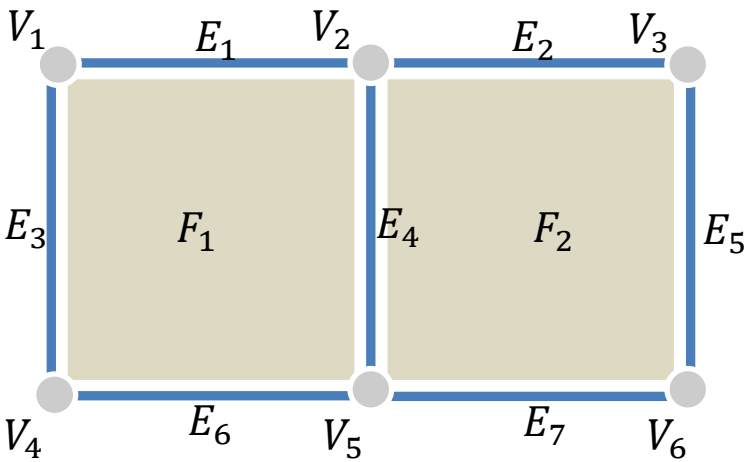
Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell

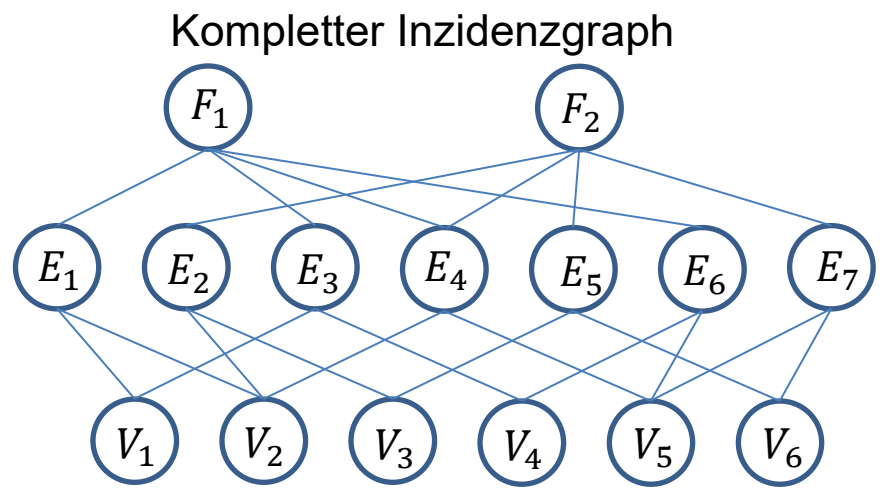


$\{V_5, E_4, F_1\}$



Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell





⇒ **alternatives Datenmodell für topologische Beziehungen**

Weitere Datenmodelle

Relationales Datenmodell:

- Entitäten als auch deren Beziehungen werden in Tabellen (sogenannten *Relationen*) hinterlegt
- Jede Zeile der Relation entspricht einem *Tupel*, jede Spalte einem *Attribut*
- Attribute, die zur Identitfizierung einzelner Tupel dienen, heissen *Schlüssel-Attribute (keys);* jedes Tupel einer Relation ist eindeutig durch einen *Primärschlüssel* identifiziert
- *Fremdschlüssel* verweisen auf Tupel einer anderen Relation und können mehrfach auftauchen
- Tupel dürfen sich in einer Relation *nicht wiederholen*. Bedeutung einer Relation darf *nicht von der Reihenfolge* der Attribute oder Tupel *abhängen*.

Weitere Datenmodelle

Relationales Datenmodell:

- Entitäten als auch deren Beziehungen werden in Tabellen (sogenannten *Relationen*) hinterlegt
- Jede Zeile der Relation entspricht einem *Tupel*, jede Spalte einem *Attribut*
- Attribute, die zur Identifizierung einzelner Tupel dienen, heissen *Schlüssel-Attribute (keys);*
- hrfach
- f *nicht*

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

Weitere Datenmodelle

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden warden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:

- Numerische Codes für textbasierte Attribute einführen

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
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 – 298.9 MJ

Weitere Datenmodelle

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden warden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:

- Numerische Codes für textbasierte Attribute einführen

- **1. Normalform**:
  - alle Attributwerte müssen *atomar* sein, d.h. sie bestehen nur aus einem einzigen Wert
  - die Relation muss *frei von Wiederholungsgruppen* sein, d.h. verschiedene Attribute dürfen keine gleiche oder gleichartige Bedeutung haben (numerische Codierung ausgenommen)

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
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ

Weitere Datenmodelle

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden warden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:

- Numerische Codes für textbasierte Attribute einführen

- **1. Normalform**:
  - alle Attributwerte müssen *atomar* sein, d.h. sie bestehen nur aus einem einzigen Wert
  - die Relation muss *frei von Wiederholungsgruppen* sein, d.h. verschiedene Attribute dürfen keine gleiche oder gleichartige Bedeutung haben (numerische Codierung ausgenommen)

- **2. Normalform**:
  - die Relation liegt bereits in 1. Normalform vor
  - jedes Nicht-Schlüsselattribut muss von *allen* Primärschlüsseln abhängen (voll funktional abhängig)

Weitere Datenmodelle

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
1 ~~2~~	3	Grit	6	Sandstein	5	Oberkarbon	323.2 MJ	
3	4	Slab		Schiefer	5	Oberkarbon		
4	1	Mount						
5	1	Mount Fm.	2					66 MJ
6	5	Volcano Fm.						201.3 MJ
7	1	Mou			8	Kreide		
8	2	Shell		Kalkstein	5	Oberkarbon	323.2 MJ	
9	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ

Jetzt sind Polygon *und* Formation Primärschlüssel, aber nicht alle Attribute sind voll funktional abhängig von den Primärschlüsseln!

Es müsste eine neue Relation angelegt warden, die nur Formation# als Primärschlüssel hat.

- die Relation muss *frei von Wiederholungsgruppen* sein, d.h. verschiedene Attribute dürfen keine gleiche oder gleichartige Bedeutung haben (numerische Codierung ausgenommen)

- **2. Normalform**:
  - die Relation liegt bereits in 1. Normalform vor
  - jedes Nicht-Schlüsselattribut muss von *allen* Primärschlüsseln abhängen (voll funktional abhängig)

Weitere Datenmodelle

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden warden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:

- **3. Normalform**:
  - die Relation liegt bereits in 2. Normalform vor
  - Alle Nicht-Schlüsselattribut sind untereinander unabhängig (d.h. sie müssen *nicht-transitiv* bzgl. dem Primärschlüssel sein)



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
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ

Formation ist nicht-transitiv von *Polygon* abhängig, aber *Lithologie* nicht ⇒ Relation aufteilen

Weitere Datenmodelle

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

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

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

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

Weitere Datenmodelle

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden warden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:

- **3. Normalform**:
  - die Relation liegt bereits in 2. Normalform vor
  - Alle Nicht-Schlüsselattribut sind untereinander unabhängig (d.h. sie müssen *nicht-transitiv* bzgl. dem Primärschlüssel sein)

- Weiter Normalformen können hinzugefügt warden, aber die 3. Normalform reicht aus, um sinnvoll Operationen der relationalen Algebra wie **Vereinigungen**, **Differenzen**, **Durchschnitte**, **karthesische Produkte**, **Projektionen**, **Selektion** von Relationen zu definieren.

- Änderungen und Ergänzungen können ebenfalls einfach vorgenommen werden

Weitere Datenmodelle

Table 2-6. Editing the relation shown in Table 2-4 B before normalization. It has been discovered that the Mount Fm. is a gabbro not a granite (!). Even in this very trivial example it can be seen that the change must be repeated exactly the same in three places in the unnormalized table. Change lithology # and lithology each time the Mount Fm. occurs, see items in bold italics.

Polygon #	Formation #	Formation Name	Lithology #	Lithology	Age #	Age
1	2	Shelly Fm.	7	Limestone	5	Pennsylvanian
2	3	Grit Fm.	6	Sandstone	5	Pennsylvanian
3	4	Slab Fm.	5	Shale	5	Pennsylvanian
4	1	Mount Fm.	***4***	***Gabbro***	8	Cretaceous
5	1	Mount Fm.	***4***	***Gabbro***	8	Cretaceous
6	5	Volcano Fm.	3	Tuff	7	Triassic
7	1	Mount Fm.	***4***	***Gabbro***	8	Cretaceous
8	2	Shelly Fm.	7	Limestone	5	Pennsylvanian
9	4	Slab Fm.	5	Shale	1	Pennsylvanian
10	2	Shelly Fm.	7	Limestone	5	Pennsylvanian

Table 2-7. After normalization, the POLYGON and AGE relations remain unchanged, and two small changes are needed in the FORMATION and LITHOLOGY relations. Suppose this change is made in a real situation for a geological map containing several thousand polygons. The editing changes would still be the same as shown here, because the POLYGON relation requires no modification.

FORMATION relation (1 change shown in italics)

Formation #	Formation Name	Lithology #	Age #
1	Mount Fm.	***4***	8
2	Shelly Fm.	7	5
3	Grit Fm.	6	5
4	Slab Fm.	5	5
5	Volcano Fm.	3	7

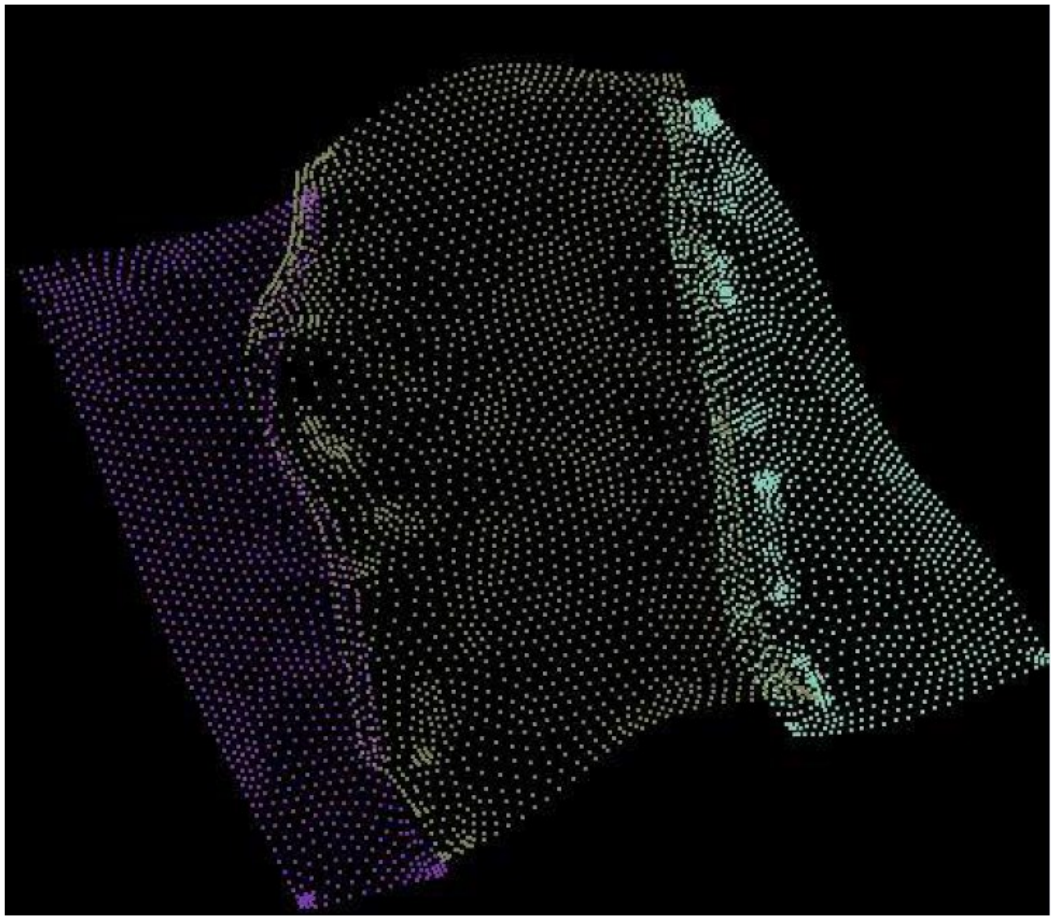
LITHOLOGY relation
(1 insertion shown in italics)

Lithology #	Lithology
2	Granite
3	Tuff
4	***Gabbro***
5	Shale
6	Sandstone
7	Limestone

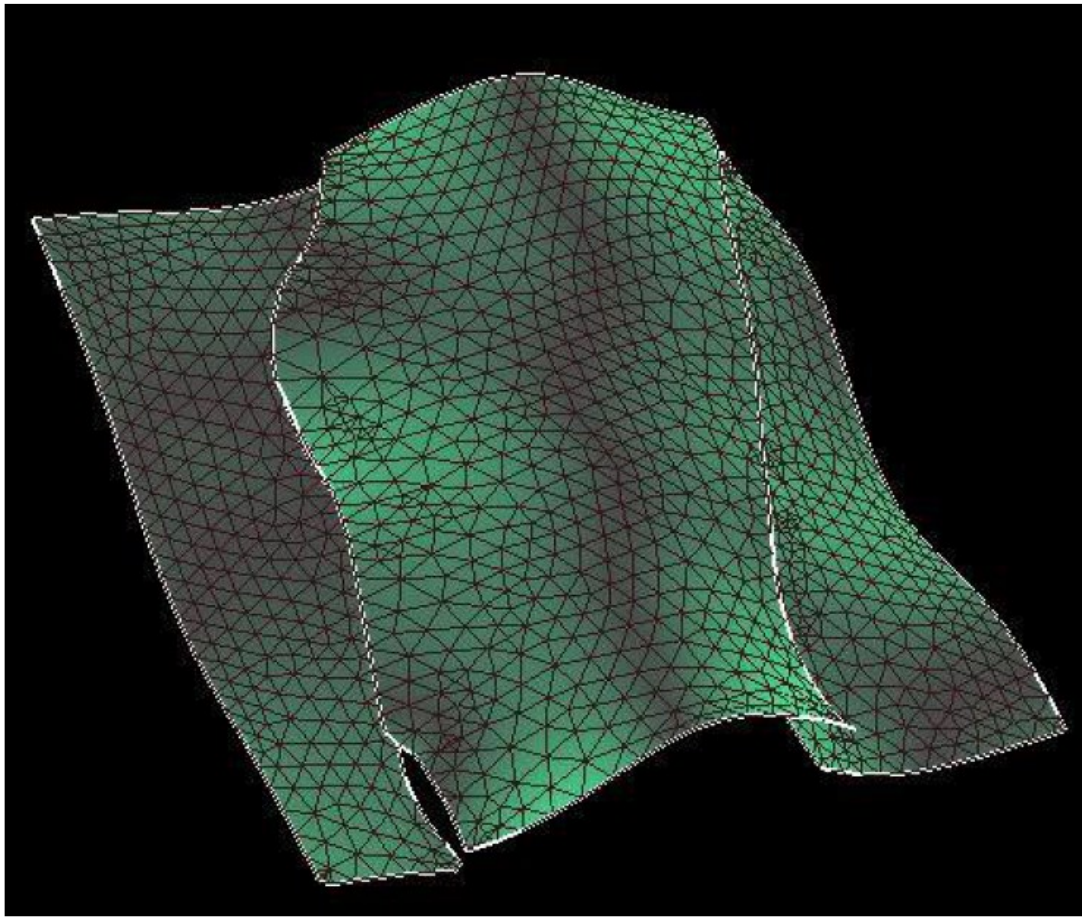
Übersicht

1. Definitionen, Funktionen, Anwendungen (Vorlesung 1)
2. Koordinatensysteme und -transformationen (Vorlesung 2+3)
3. Räumliche Datenmodellierung (Vorlesung 4 - 6)
4. **Vermaschungen**
5. Räumliche Interpolation
6. Transformationen, Filtermethoden, Sonstiges

Zerlegungen und Triangulierung (Partition and Triangulation)

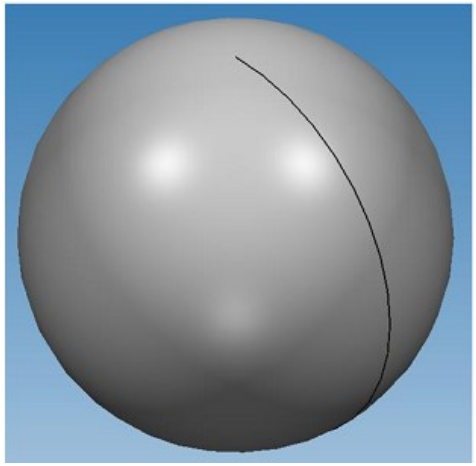


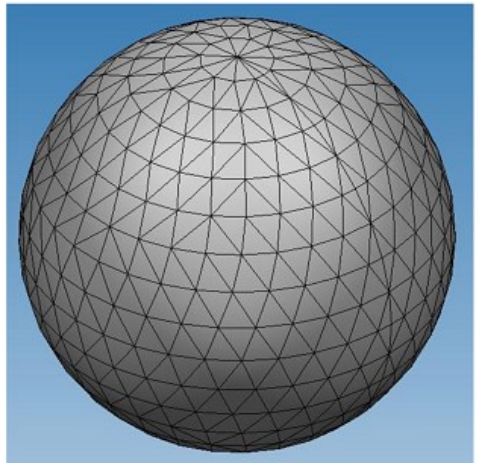
Zerlegungen und Triangulierung (Partition and Triangulation)



Zerlegungen und Triangulierung (Partition and Triangulation)

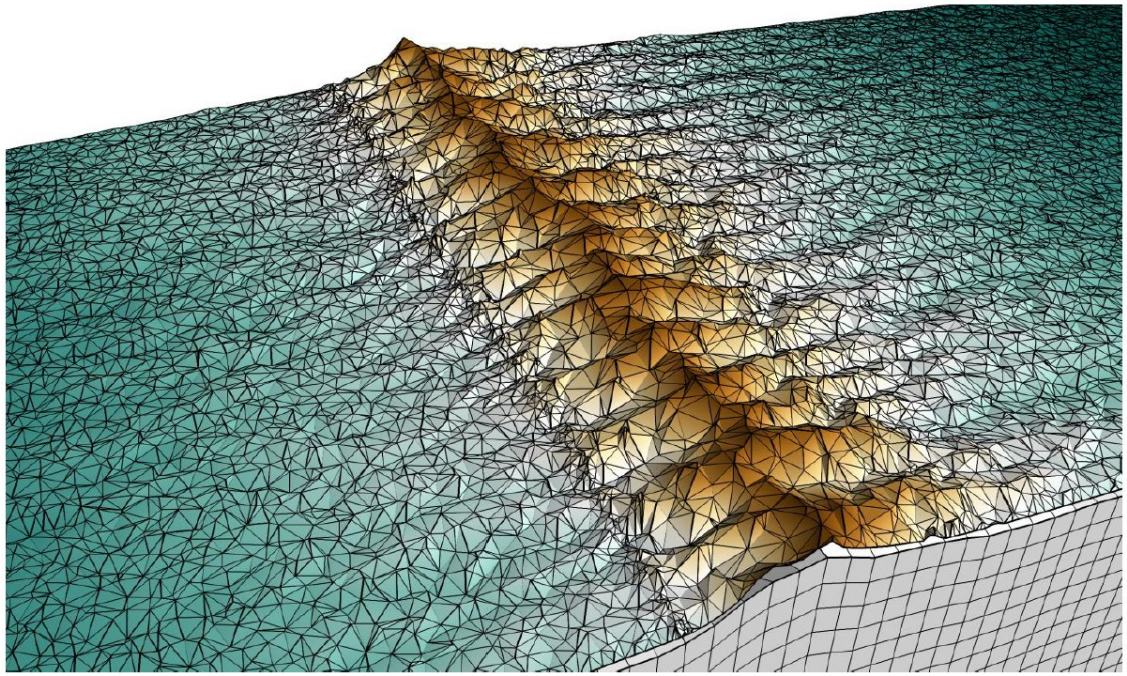
Im Bereich “computer aided geometric design” oder “computational geometry” bezieht sich der Terminus “Zerlegung” auf jedes beliebige Verfahren, um ein beliebiges n-dimensionales Objekt in eine Menge benachbarter polytopaler* n-dimensionaler Zellen zu zerlegen.





Zerlegungen und Triangulierung (Partition and Triangulation)

Im Kontext von GIS werden Zerlegungen verwendet, wenn ein Geoobjekt über seine Bestandteile beschrieben werden soll. Diesen Bestandteilen können dadurch eigene Attribute zugewiesen werden.



Zerlegungen und Triangulierung (Partition and Triangulation)

Im Kontext von GIS werden Zerlegungen verwendet, wenn ein beliebiges Geoobjekt über seine Bestandteile beschrieben werden soll. Diesen Bestandteilen können dadurch eigene Attribute zugewiesen werden.

Auf diesen Bestandteilen lassen sich Berechnungen zumeist sehr viel einfacher durchführen, als auf dem Objekten selbst. Eine digitale visuelle Darstellung eines beliebigen Objektes ist ohne eine Vermaschung des Objektes in einfache diskrete Zellen oft nicht möglich.

Zerlegungen sind eine spezielle Form des Vektormodells.

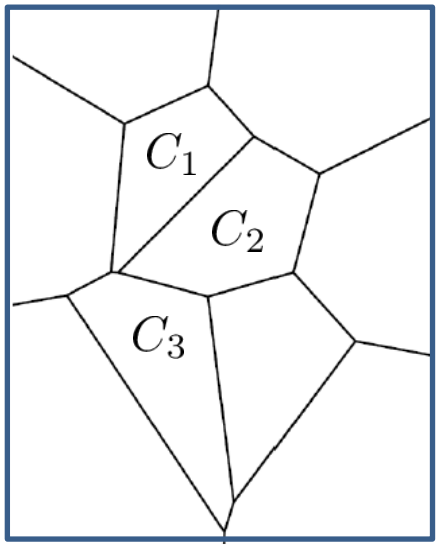
Zerlegung / Vermaschung

Eine **Zerlegung** (*partition*) einer Region $\Omega \subset \mathbb{R}^n$ ist eine Menge offener Zellen $C_i \subset \mathbb{R}^n, i = 1, \dots, N$ mit $C_i \cap C_j = \emptyset, i \neq j$, so dass gilt

$$\Omega = \bigcup_{i=1,\dots,N} C_i$$

Eine solche Zerlegung wird auch als Vermaschung (*tesselation*) bezeichnet.

Die „*Offenheit*“ der Zellen ist für eine exakte mathematische Beschreibung notwendig …



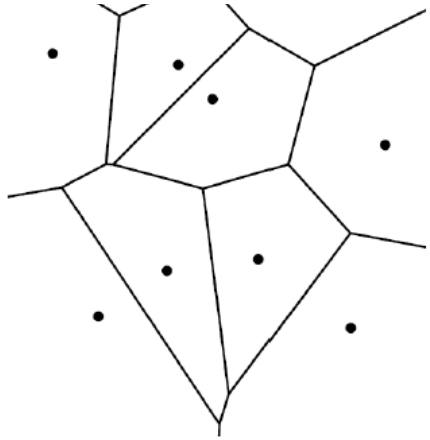
Voronoi Vermaschung

Gegeben sei eine Punktmenge $P = \{p_i \in \mathbb{R}^n : i = 1, \dots N\}$. Eine **Voronoi-Zelle V_i** beschriebt die Menge

$$V_i = \{x \in \mathbb{R}^n : |x - p_i| < |x - p_j|, i \neq j\}.$$

Eine solche Zelle ist eine spezielle Form eines Polytops (z.B. ebener Polyeder in 3D od. ebenes Polygon in 2D). Dabei bilden die Zellen $C_i = V_i$ „annähernd“* eine Zerlegung des $\mathbb{R}^n$.

*Warum nur „annähernd“?
Formal müssten die M Grenzkanten aller Zellen mit als zusätzliche Zellen $C_i, i = N + 1, \dots, N + M$ und alle K Knoten, an denen sich Kanten treffen, als zusätzliche Zellen $C_i, i = N + M + 1, \dots, N + M + K$ mit einbezogen werden.
Diese Zellen werden durch die „Offenheits“-Bedingung nicht automatisch erfasst.



Voronoi Vermaschung

Gegeben sei eine Punktmenge $P = \{p_i \in \mathbb{R}^n : i = 1, \dots N\}$. Eine **Voronoi-Zelle V_i** beschriebt die Menge

$$V_i = \{x \in \mathbb{R}^n : |x - p_i| < |x - p_j|, i \neq j\}.$$

Eine solche Zelle ist eine spezielle Form eines Polytops (z.B. ebener Polyeder in 3D od. ebenes Polygon in 2D). Dabei bilden die Zellen $C_i = V_i$ „annähernd“* eine Zerlegung des $\mathbb{R}^n$.

Der **Bisector** zwischen zwei Punkten p_i, p_j ist gegeben durch

$$B_{i,j} = \{x \in \mathbb{R}^n : |x - p_i| = |x - p_j|, i \neq j\}$$

Und teilt den $\mathbb{R}^n$ in zwei Halbräume. $B_{i,j}$ repräsentiert die Grenze des Halbraums

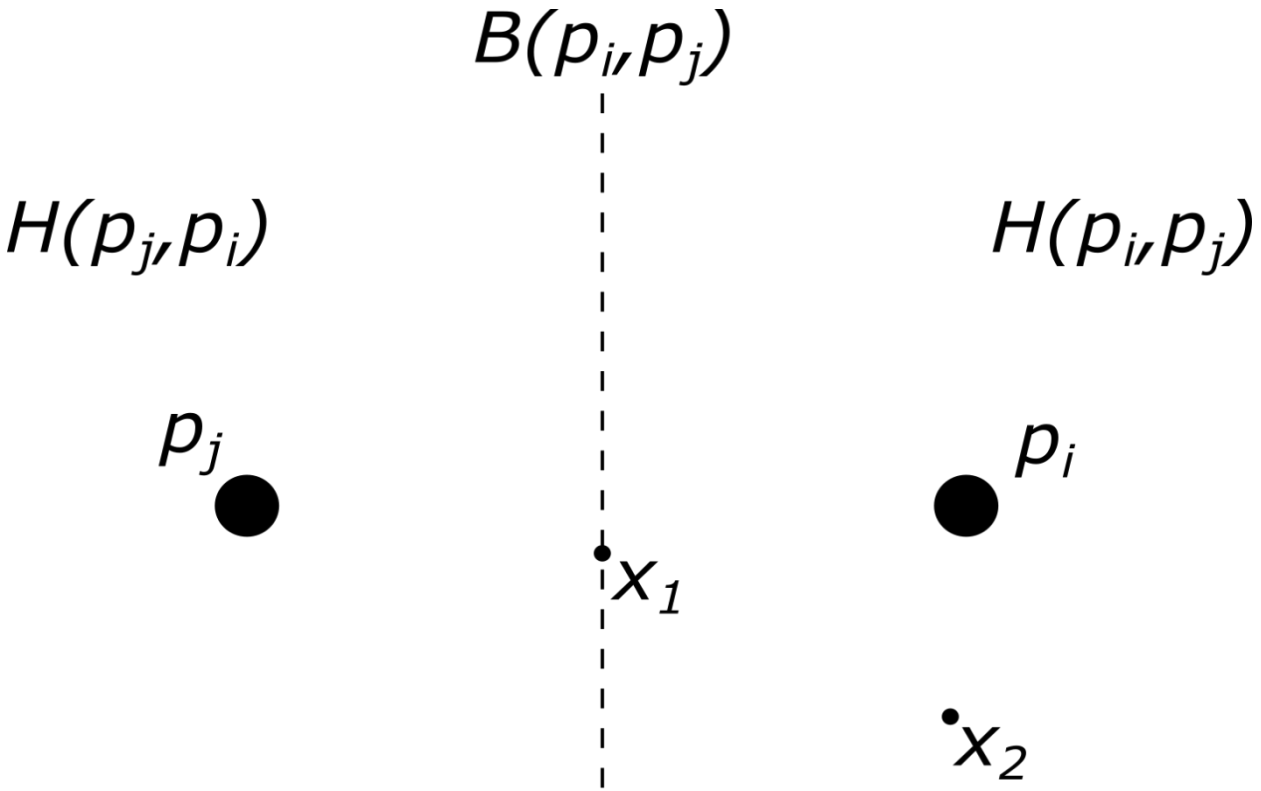
$$H_{i,j} = \{x \in \mathbb{R}^n : |x - p_i| < |x - p_j|, i \neq j\}.$$

Jede Voronoi-Zelle kann über die Schnittmenge

$$V_i = \bigcup_{j=1,\dots,N; i \neq j} H_{i,j}$$

repräsentiert werden.

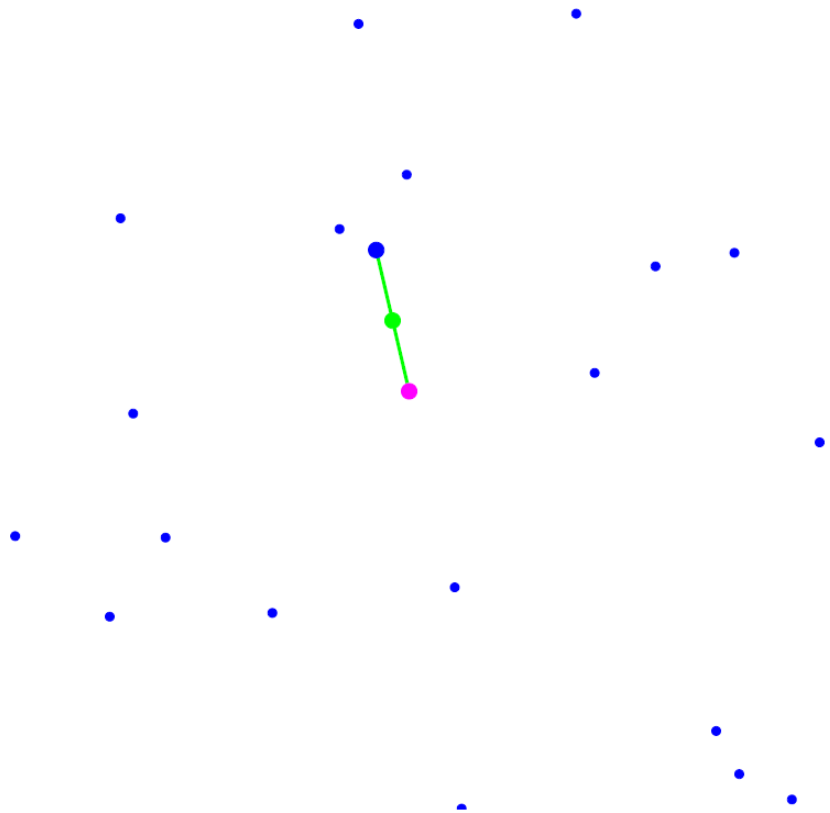
Voronoi Vermaschung



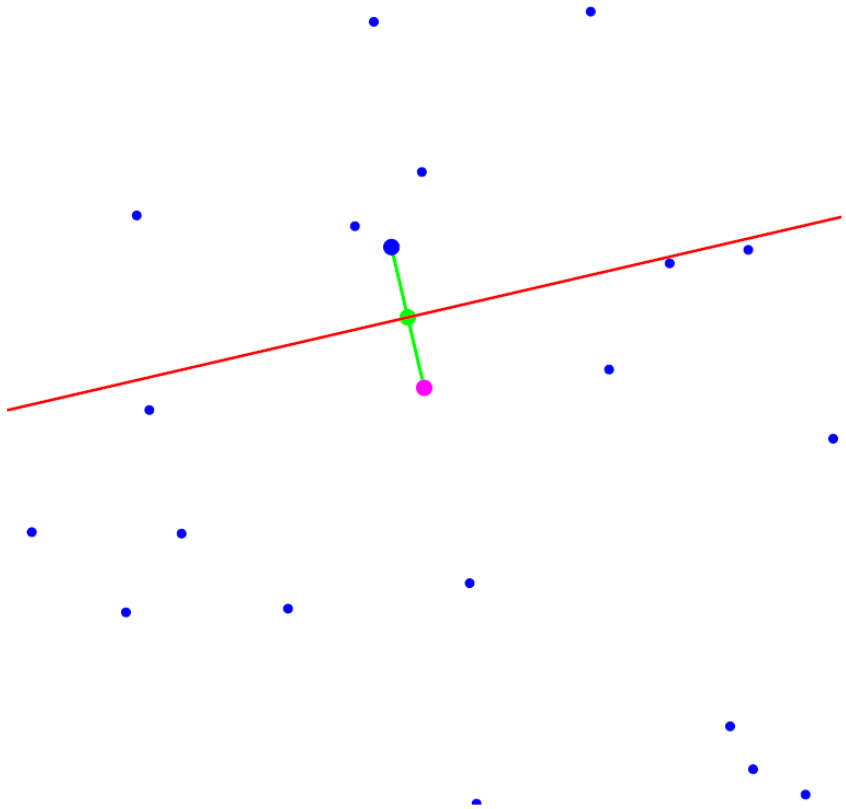
$$|x_1 - p_i| = |x_1 - p_j| \rightarrow x_1 \in B(p_i, p_j)$$

$$|x_2 - p_i| < |x_2 - p_j| \rightarrow x_2 \in H(p_i, p_j)$$

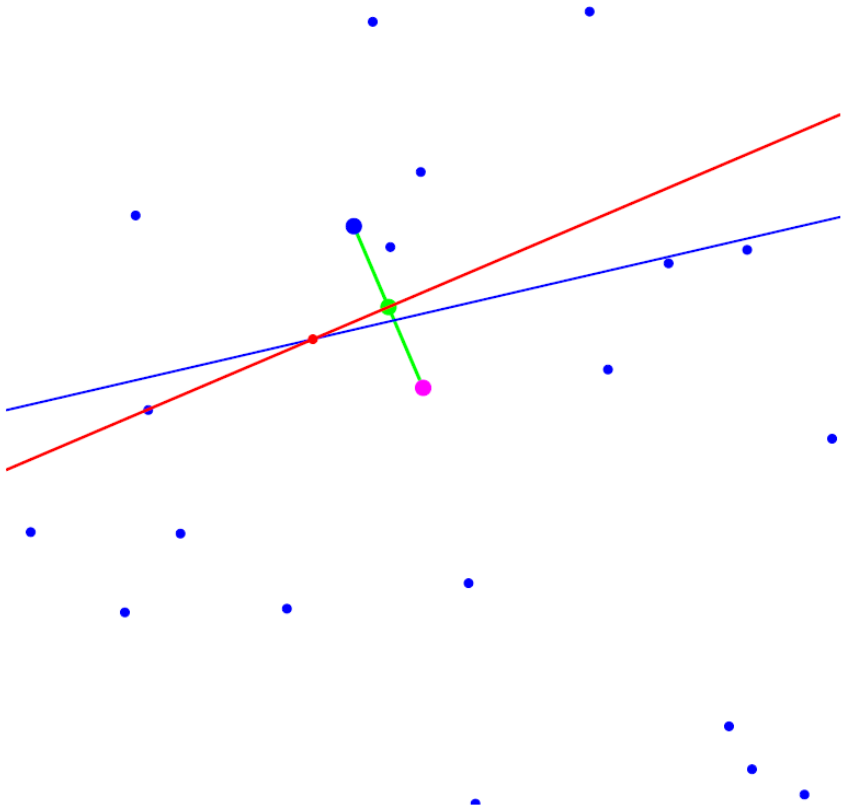
Beispiel: Voronoi Vermaschung in 2-D



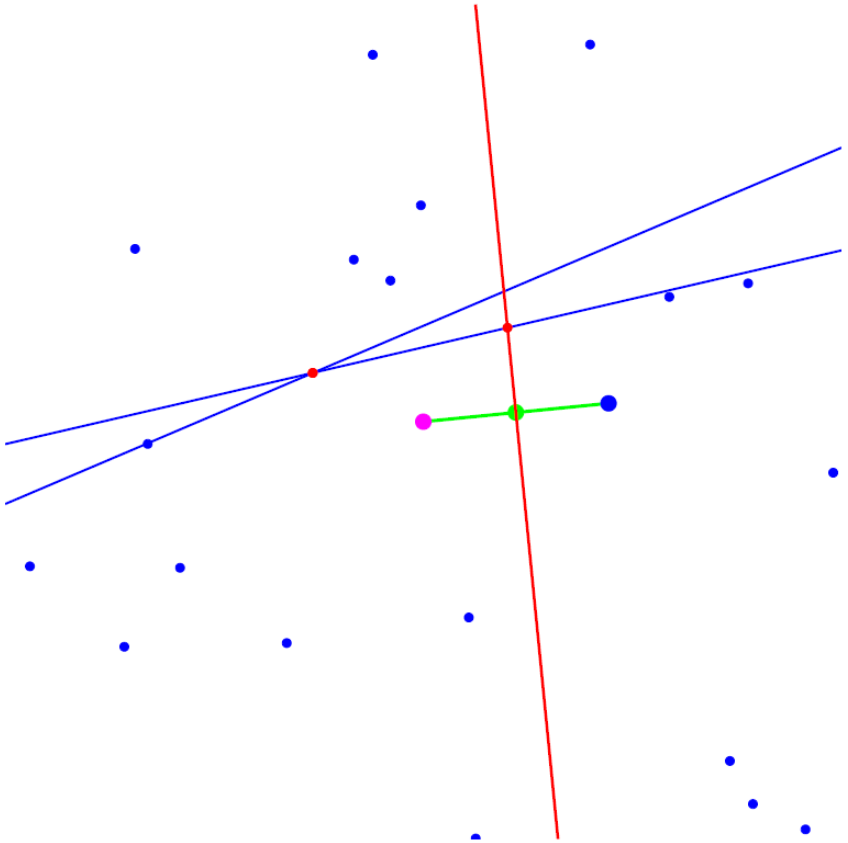
Beispiel: Voronoi Vermaschung in 2-D



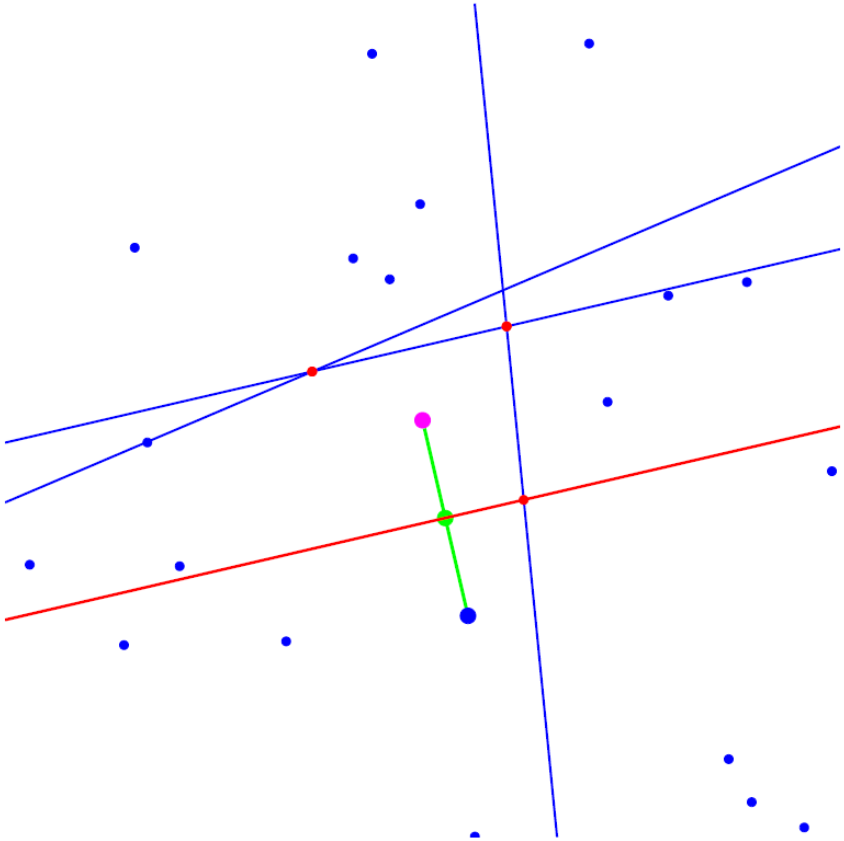
Beispiel: Voronoi Vermaschung in 2-D



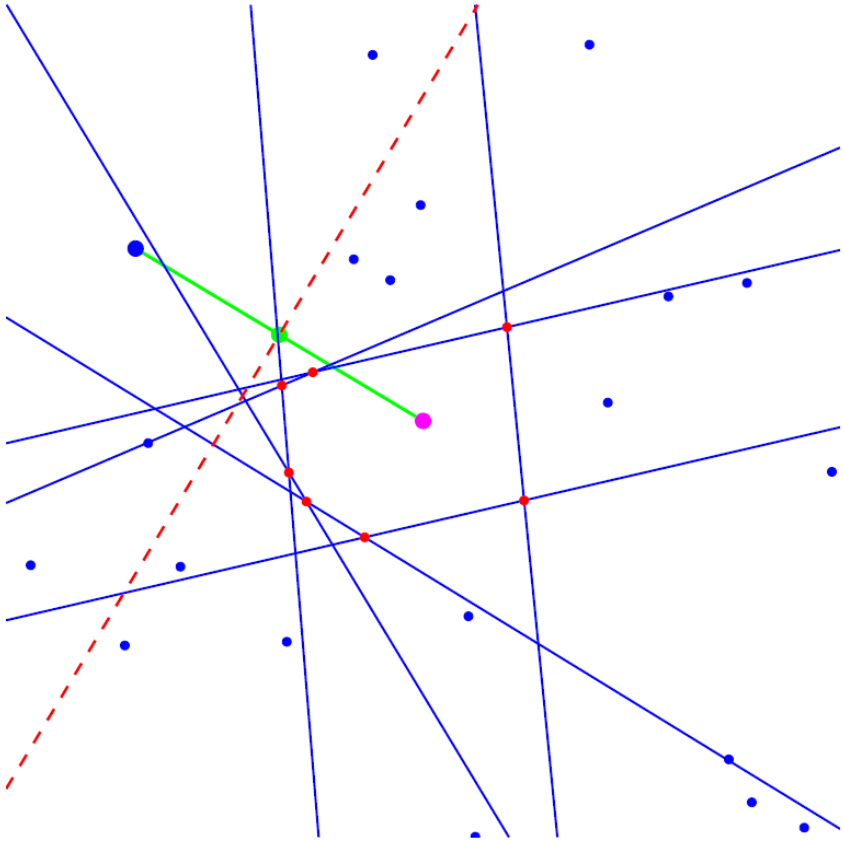
Beispiel: Voronoi Vermaschung in 2-D



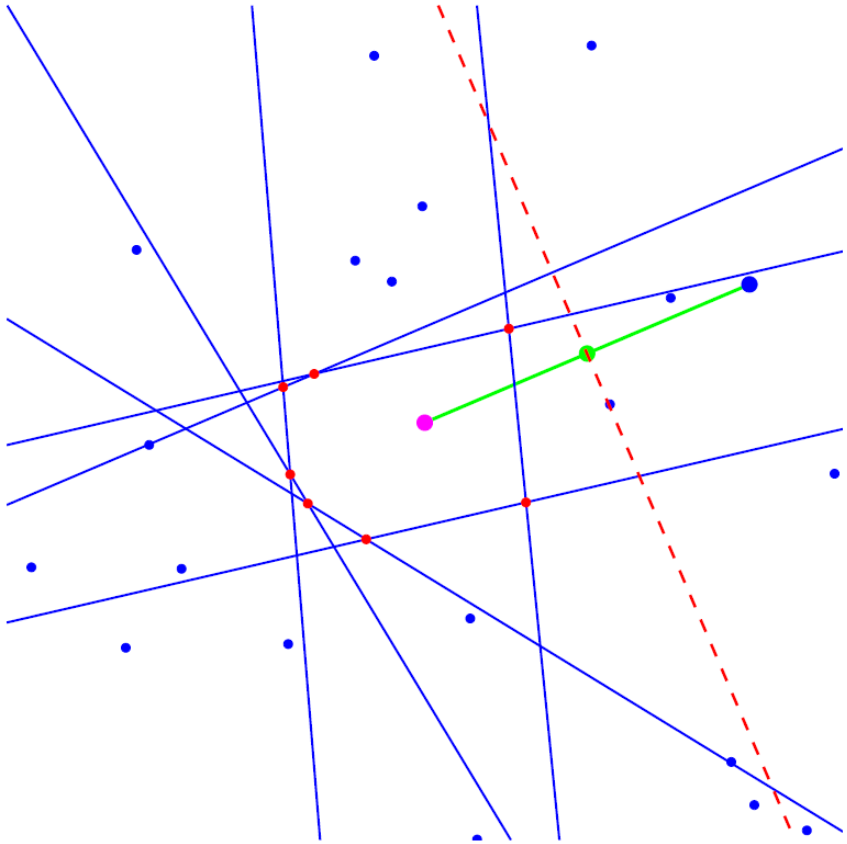
Beispiel: Voronoi Vermaschung in 2-D



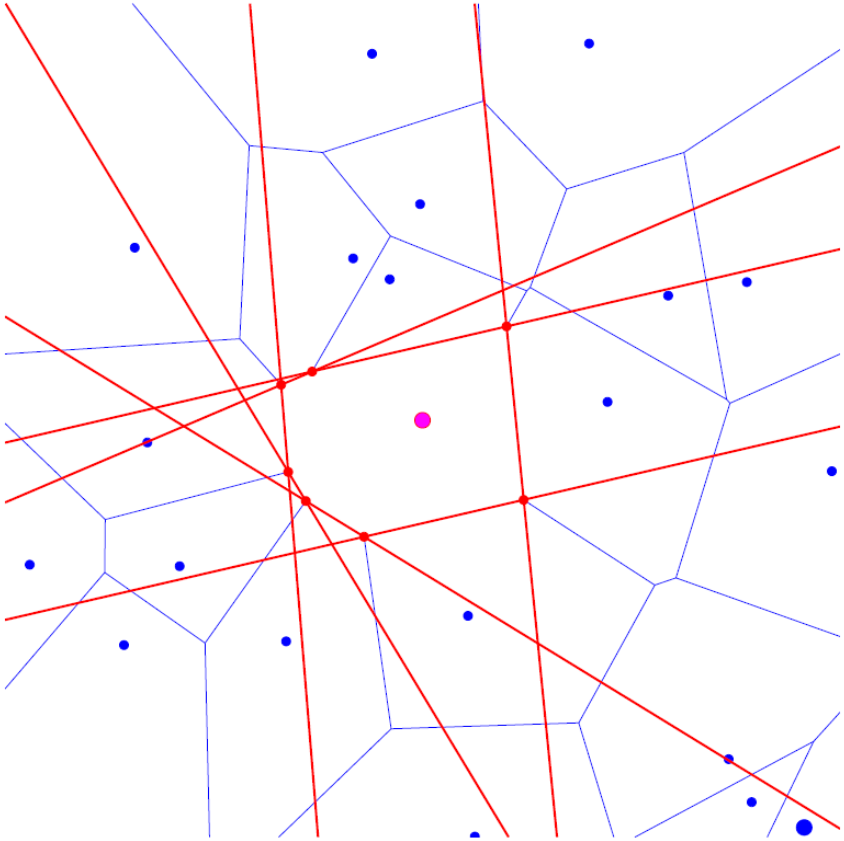
Beispiel: Voronoi Vermaschung in 2-D



Beispiel: Voronoi Vermaschung in 2-D



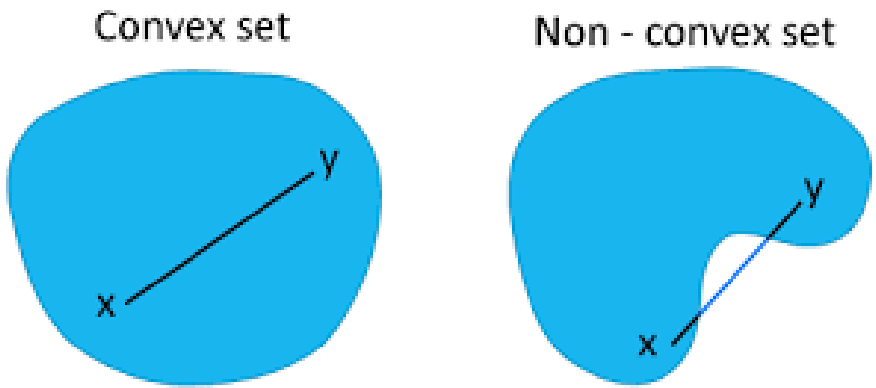
Beispiel: Voronoi Vermaschung in 2-D



Voronoi Vermaschung

Lemma (Eigenschaften). Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ ist die Voronoi-Vermaschung der Punktmenge $P = \{P_1, \dots, P_N\} \subset \mathbb{R}^{n*}$.
Dann gelten folgende Eigenschaften:

a) Jede Zelle V_i ist konvex (d.h. wenn V_i zwei Punkte x, y enthält, muss auch die komplette Strecken zwischen x und y enthalten sein).



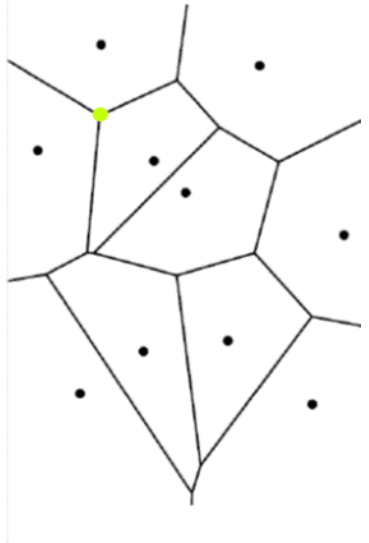
*"*in general position*": es dürfen keine $n + 1$ Punkte auf einer $n - 1$-Kugel liegen

Voronoi Vermaschung

Lemma (Eigenschaften). Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ ist die Voronoi-Vermaschung der Punktmenge $P = \{P_1, \dots, P_N\} \subset \mathbb{R}^n$.
Dann gelten folgende Eigenschaften:

a) Jede Zelle V_i ist konvex (d.h. wenn V_i zwei Punkte x, y enthält, muss auch die komplette Strecken zwischen x und y enthalten sein).

b) Jeder Vertex einer Voronoi-Vermaschung liegt auf der Grenze von exakt $\mathrm{n} + 1$ verschiedenen Zellen $V_i \in \mathcal{V}$



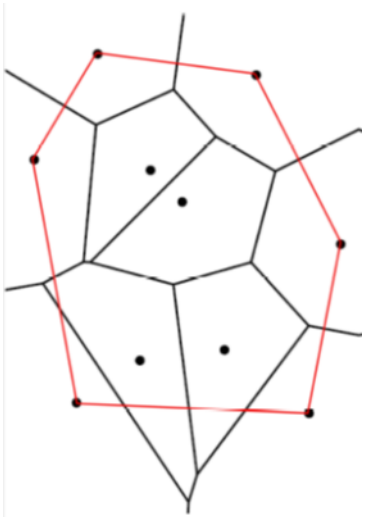
Voronoi Vermaschung

Lemma (Eigenschaften). Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ ist die Voronoi-Vermaschung der Punktmenge $P = \{P_1, \dots, P_N\} \subset \mathbb{R}^n$.
Dann gelten folgende Eigenschaften:

a) Jede Zelle V_i ist konvex (d.h. wenn V_i zwei Punkte x, y enthält, muss auch die komplette Strecken zwischen x und y enthalten sein).

b) Jeder Vertex einer Voronoi-Vermaschung liegt auf der Grenze von exakt $\mathrm{n} + 1$ verschiedenen Zellen $V_i \in \mathcal{V}$.

c) Eine Zelle V_i ist *unbegrenzt* (unendlich ausgedehnt), wenn p_i auf der konvexen Hülle $\mathcal{C}(P)$ von P liegt. Diese ist das kleinste konvexe n-Polytop, welches P umfasst.



Voronoi Vermaschung

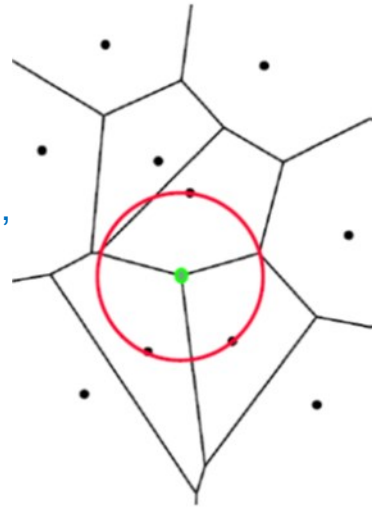
Lemma (Eigenschaften). Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ ist die Voronoi-Vermaschung der Punktmenge $P = \{p_1, \dots, p_N\} \subset \mathbb{R}^n$.
Dann gelten folgende Eigenschaften:

a) Jede Zelle V_i ist konvex (d.h. wenn V_i zwei Punkte x, y enthält, muss auch die komplette Strecken zwischen x und y enthalten sein).

b) Jeder Vertex einer Voronoi-Vermaschung liegt auf der Grenze von exakt $\mathrm{n} + 1$ verschiedenen Zellen $V_i \in \mathcal{V}$.

c) Eine Zelle V_i ist *unbegrenzt* (unendlich ausgedehnt), wenn p_i auf der konvexen Hülle $\mathcal{C}(P)$ von P liegt. Diese ist das kleinste konvexe n-Polytop, welches P umfasst.

d) Sei X ein Vertex der zu den Zellen $V_{i_1}, \dots, V_{i_{n+1}}$ gehört. Dann ist X der Mittelpunkte einer n-Kugel durch die Punkte $p_{i_1}, \dots, p_{i_{n+1}}$.
Es liegt kein anderer Punkt von P innerhalb dieser n-Kugel.



Voronoi Zerlegung – Tabellarische Darstellung 2D

Es muss in irgendeiner Form beschrieben werden, wie die Voronoizellen geometrisch definiert sind:

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

Interpolation mittels Voronoi Zerlegung: Nearest Neighbor Interpolation

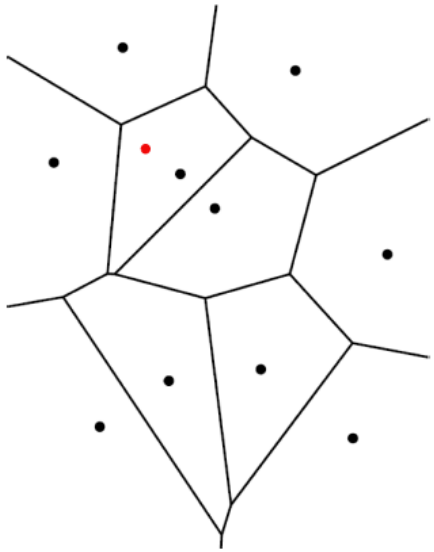
Gegen sein eine Punktmenge $P = \{p_1, \dots, p_N\} \subset \mathbb{R}^n$ und ein Messwert/Datum h_i wird jedem Punkt p_i zugewiesen.

Nearest-Neighbor-Interpolation: Gegen sein ein beliebiger Punkt $x \in \mathbb{R}^n$, an dem kein Messwert $h(x)$ bekannt ist.
Dieser kann mittels der zugehörigen Voronoi-Vermaschung $\mathcal{V}(P)$ vorhergesagt werden:

- x kann nur in einer Voronoi-Zelle liegen $x \in V_i$

➢ $h(x) = h_i$

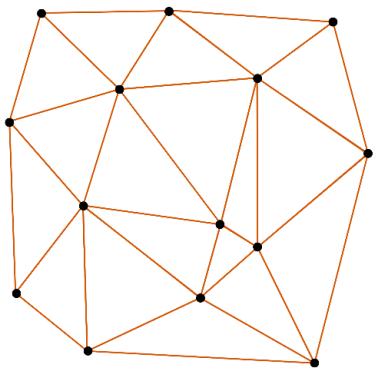
Wie gut wäre eine solche Vorhersage?



Vermaschungen als Graphen

Ein Graph $\mathcal{G} = (V, E)$ ist ein Tupel von abstrakten Knoten/Vertices $V = \{v_1, \dots, v_N\}$ und Kanten $\mathrm{E} = \{e_1, \dots, e_N\}$, von denen jede zwei Knoten verbindet. Ein Graph wird als „einfach“ (*simple*) bezeichnet, wenn Folgendes gilt:

1. Zwei Knoten werden durch maximal eine Kante verbunden,
2. Kanten haben keine Orientierung
3. Es existieren keine Schleifen (*loops*, Kanten mit identischem Start- und Endpunkt)



Vermaschungen als Graphen

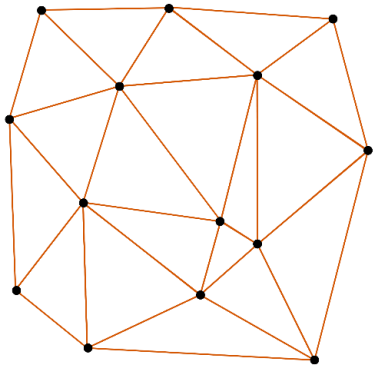
Ein Graph $\mathcal{G} = (V, E)$ ist ein Tupel von abstrakten Knoten/Vertices $V = \{v_1, \dots, v_N\}$ und Kanten $\mathrm{E} = \{e_1, \dots, e_N\}$, von denen jede zwei Knoten verbindet. Ein Graph wird als „einfach“ (*simple*) bezeichnet, wenn Folgendes gilt:

1. Zwei Knoten werden durch maximal eine Kante verbunden,

2. Kanten haben keine Orientierung

3. Es existieren keine Schleifen (*loops*, Kanten mit identischem Start- und Endpunkt)

Knoten entsprechen 0-Zellen, Kanten 1-Zellen und Flächen, begrenzt durch einen Kantenzug, 2-Zellen. Kanten bestimmen die „Nachbarschaft“ von Flächen und Knoten die Nachbarschaft von Kanten.



Vermaschungen als Graphen

Ein Graph kann mittels seiner **Adjazenz-Matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, \text{für } i \neq j \text{ und } v_i, v_j \text{ sind über eine Kante verbunden} \\ 0, \text{andernfalls} \end{cases}.$$

Eine äquivalente Charakterisierung kann über die Inzidenz-Matrix $I_{VE} \in \mathbb{R}^{N \times N}$ erfolgen:

$$(I_{VE})_{i,j} \begin{cases} 1, \text{falls Vertex } v_i \text{ ist Endpunkt von Kante } e_j \\ 0, \text{andernfalls} \end{cases}.$$

Jede Spalte in I_{VE} eines einfachen Graphen enthält den Wert 1 genau zwei mal.

Vermaschungen als Graphen

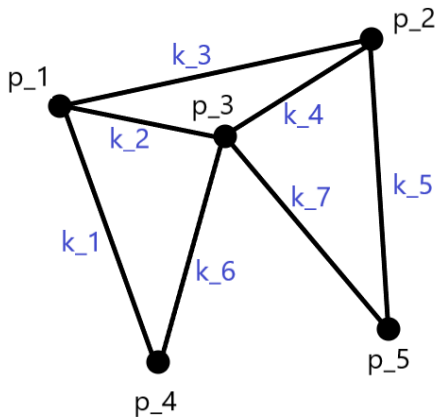
Ein Graph kann mittels seiner **Adjazenz-Matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, \text{für } i \neq j \text{ und } v_i, v_j \text{ sind über eine Kante verbunden} \\ 0, \text{andernfalls} \end{cases}.$$

Eine äquivalente Charakterisierung kann über die Inzidenz-Matrix $I_{VE} \in \mathbb{R}^{N \times N}$ erfolgen:

$$(I_{VE})_{i,j} \begin{cases} 1, \text{falls Vertex } v_i \text{ ist Endpunkt von Kante } e_j \\ 0, \text{andernfalls} \end{cases}.$$

Jede Spalte in I_{VE} eines einfachen Graphen enthält den Wert 1 genau zwei mal.



Adjazent-Matrix dieses Beispiel-Graphen:

$$A_V = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Vermaschungen als Graphen

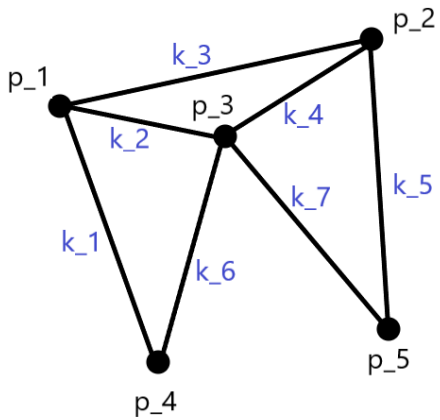
Ein Graph kann mittels seiner **Adjazenz-Matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, \text{für } i \neq j \text{ und } v_i, v_j \text{ sind über eine Kante verbunden} \\ 0, \text{andernfalls} \end{cases}.$$

Eine äquivalente Charakterisierung kann über die Inzidenz-Matrix $I_{VE} \in \mathbb{R}^{N \times N}$ erfolgen:

$$(I_{VE})_{i,j} \begin{cases} 1, \text{falls Vertex } v_i \text{ ist Endpunkt von Kante } e_j \\ 0, \text{andernfalls} \end{cases}.$$

Jede Spalte in I_{VE} eines einfachen Graphen enthält den Wert 1 genau zwei mal.



Inzidenz-Matrix dieses Beispiel-Graphen:

$$I_{VE} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Vermaschungen als Graphen

Ein Graph kann mittels seiner **Adjazenz-Matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, \text{für } i \neq j \text{ und } v_i, v_j \text{ sind über eine Kante verbunden} \\ 0, \text{andernfalls} \end{cases}.$$

Eine äquivalente Charakterisierung kann über die Inzidenz-Matrix $I_{VE} \in \mathbb{R}^{N \times N}$ erfolgen:

$$(I_{VE})_{i,j} \begin{cases} 1, \text{falls Vertex } v_i \text{ ist Endpunkt von Kante } e_j \\ 0, \text{andernfalls} \end{cases}.$$

Jede Spalte in I_{VE} eines einfachen Graphen enthält den Wert 1 genau zwei mal.

Es gilt:

$$(A_V)_{i,j} = \begin{cases} (I_{VE} I_{VE}^T)_{i,j} \text{ für } i \neq \text{j} \\ 0, \text{sonst} \end{cases}$$

Topologie einer Voronoi Vermaschung

Neben Kanten und Vertices, beinhaltet einen 2-D Voronoi-Vermaschung auch die Flächen $F_1, \dots, F_K$. Die zugehörigen Inzidenz-Matrizen können wie folgt repräsentiert werden:

$$(\mathrm{I}_{VE})_{i,j} = \begin{cases} 1, \text{falls } v_i \text{ ist Endpunkt von } e_j \\ 0, \text{sonst} \end{cases}$$

$$(\mathrm{I}_{EF})_{i,j} = \begin{cases} 1, \text{falls } e_i \text{ ist Grenzkante von } F_j \\ 0, \text{sonst} \end{cases}$$

$$(\mathrm{I}_{VF})_{i,j} = (\mathrm{I}_{VE}\mathrm{I}_{EF})_{i,j} = \begin{cases} 1, \text{falls } v_i \text{ ist Grenzknoten von } F_j \\ 0, \text{sonst} \end{cases}$$

Topologie einer Voronoi Vermaschung

Neben Kanten und Vertices, beinhaltet einen 2-D Voronoi-Vermaschung auch die Flächen $F_1, \dots, F_K$. Die zugehörigen Inzidenz-Matrizen können wie folgt repräsentiert werden:

$$(\mathrm{I}_{VE})_{i,j} = \begin{cases} 1, \text{falls } v_i \text{ ist Endpunkt von } e_j \\ 0, \text{sonst} \end{cases}$$

$$(\mathrm{I}_{EF})_{i,j} = \begin{cases} 1, \text{falls } e_i \text{ ist Grenzkante von } F_j \\ 0, \text{sonst} \end{cases}$$

$$(\mathrm{I}_{VF})_{i,j} = (\mathrm{I}_{VE}\mathrm{I}_{EF})_{i,j} = \begin{cases} 1, \text{falls } v_i \text{ ist Grenzknoten von } F_j \\ 0, \text{sonst} \end{cases}$$

Diese Matrizen beschreiben die Nachbarschaftsbeziehungen zwischen Vertices, Kanten und Flächen. So lässt sich die Nachbarschaft zweier Voronoi-Zellen (Flächen) über folgende Matrix beschreiben:

$$(A_F)_{i,j} = \begin{cases} (I_{EF}^T I_{EF})_{i,j}, \text{falls } i \neq \mathrm{j} \\ 0, \text{sonst} \end{cases}$$

Der Duale Graph

Der **duale Graph** $\mathcal{G}^* = (V^*, E^*)$ zum Graph $\mathcal{G}$ ordnet die Vertices V^* den Flächen F aus $\mathcal{G}$ zu. Die Kantenmenge E^* beinhaltet Kanten zwischen zwei Vertices aus V^* wenn die zugehörigen Flächen aus $\mathcal{G}$ zueinander adjazent sind. Aus diesem Grund sind die Adjazent-Matrizen A_{V^*} und A_F identisch.

Der Duale Graph

Der **duale Graph** $\mathcal{G}^* = (V^*, E^*)$ zum Graph $\mathcal{G}$ ordnet die Vertices V^* den Flächen F aus $\mathcal{G}$ zu. Die Kantenmenge E^* beinhaltet Kanten zwischen zwei Vertices aus V^* wenn die zugehörigen Flächen aus $\mathcal{G}$ zueinander adjazent sind. Aus diesem Grund sind die Adjazent-Matrizen A_{V^*} und A_F identisch.

Allgemeiner: Die duale Vermaschung einer n-dimensionalen Vermaschung ist dadurch definiert, dass ihre i-Zellen den $(n-i)$-Zellen aus der ursprünglichen Vermaschung zugeordnet werden.

Zum Beispiel wird im 2D jedem ursprünglichen Knoten (Dim. 0) eine duale Fläche (Dim. 2) zugeordnet: $\forall v_i \in V \rightarrow \exists f_i^* \in F^*$

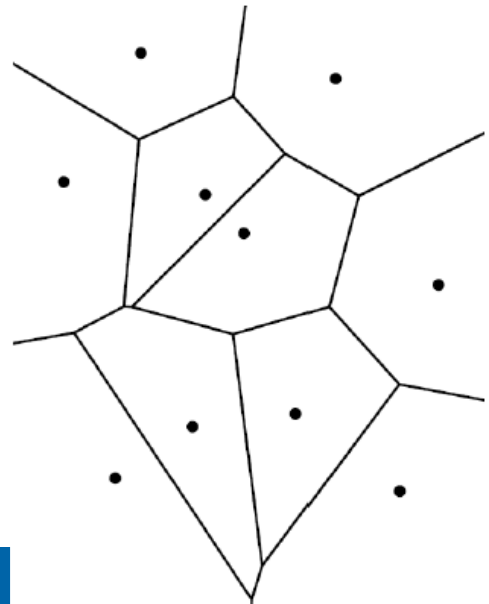
Der Duale Graph

Der **duale Graph** $\mathcal{G}^* = (V^*, E^*)$ zum Graph $\mathcal{G}$ ordnet die Vertices V^* den Flächen F aus $\mathcal{G}$ zu. Die Kantenmenge E^* beinhaltet Kanten zwischen zwei Vertices aus V^* wenn die zugehörigen Flächen aus $\mathcal{G}$ zueinander adjazent sind. Aus diesem Grund sind die Adjazent-Matrizen A_{V^*} und A_F identisch.

Allgemeiner: Die duale Vermaschung einer n-dimensionalen Vermaschung ist dadurch definiert, dass ihre i-Zellen den $(n-i)$-Zellen aus der ursprünglichen Vermaschung zugeordnet werden.

Zum Beispiel wird im 2D jedem ursprünglichen Knoten (Dim. 0) eine duale Fläche (Dim. 2) zugeordnet: $\forall v_i \in V \rightarrow \exists f_i^* \in F^*$

Eine solche duale Betrachtung einer Voronoi-Vermaschung führt zur **Delaunay Triangulierung.**



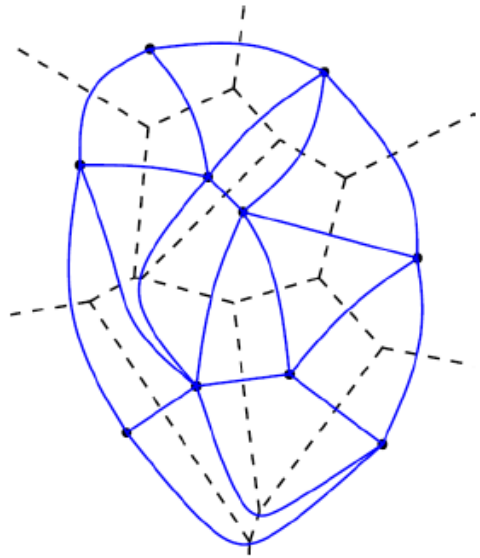
Der Duale Graph

Der **duale Graph** $\mathcal{G}^* = (V^*, E^*)$ zum Graph $\mathcal{G}$ ordnet die Vertices V^* den Flächen F aus $\mathcal{G}$ zu. Die Kantenmenge E^* beinhaltet Kanten zwischen zwei Vertices aus V^* wenn die zugehörigen Flächen aus $\mathcal{G}$ zueinander adjazent sind. Aus diesem Grund sind die Adjazent-Matrizen A_{V^*} und A_F identisch.

Allgemeiner: Die duale Vermaschung einer n-dimensionalen Vermaschung ist dadurch definiert, dass ihre i-Zellen den $(n-i)$-Zellen aus der ursprünglichen Vermaschung zugeordnet werden.

Zum Beispiel wird im 2D jedem ursprünglichen Knoten (Dim. 0) eine duale Fläche (Dim. 2) zugeordnet: $\forall v_i \in V \rightarrow \exists f_i^* \in F^*$

Eine solche duale Betrachtung einer Voronoi-Vermaschung führt zur **Delaunay Triangulierung.**



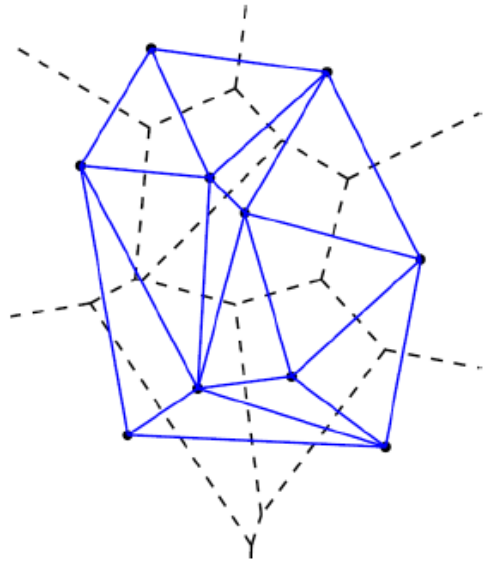
Der Duale Graph

Der **duale Graph** $\mathcal{G}^* = (V^*, E^*)$ zum Graph $\mathcal{G}$ ordnet die Vertices V^* den Flächen F aus $\mathcal{G}$ zu. Die Kantenmenge E^* beinhaltet Kanten zwischen zwei Vertices aus V^* wenn die zugehörigen Flächen aus $\mathcal{G}$ zueinander adjazent sind. Aus diesem Grund sind die Adjazent-Matrizen A_{V^*} und A_F identisch.

Allgemeiner: Die duale Vermaschung einer n-dimensionalen Vermaschung ist dadurch definiert, dass ihre i-Zellen den $(n-i)$-Zellen aus der ursprünglichen Vermaschung zugeordnet werden.

Zum Beispiel wird im 2D jedem ursprünglichen Knoten (Dim. 0) eine duale Fläche (Dim. 2) zugeordnet: $\forall v_i \in V \rightarrow \exists f_i^* \in F^*$

Eine solche duale Betrachtung einer Voronoi-Vermaschung führt zur **Delaunay Triangulierung.**

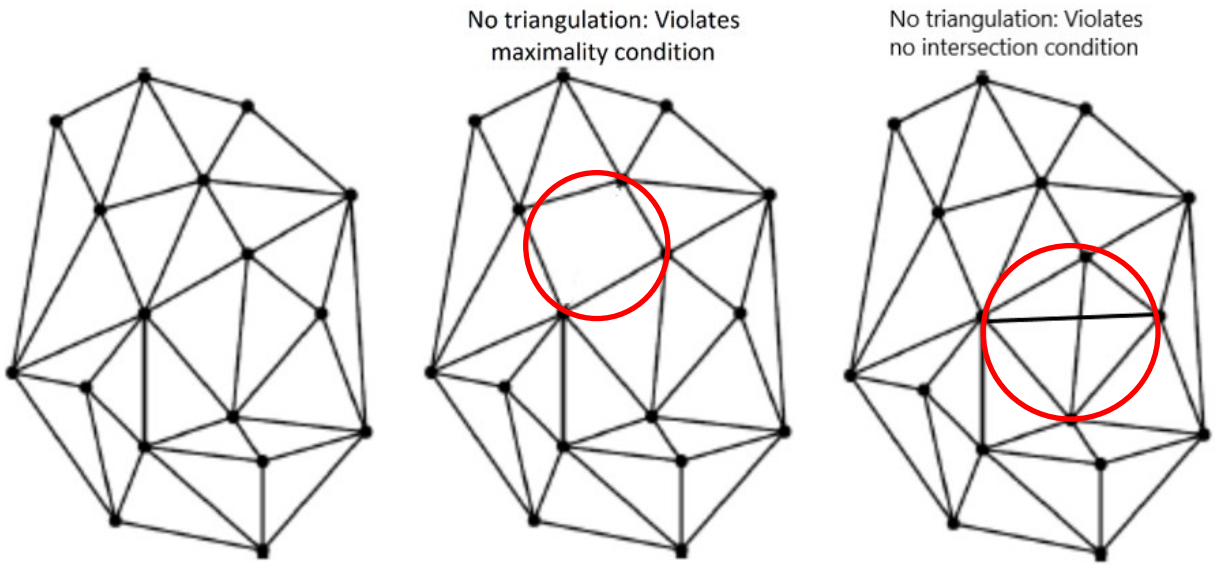


Triangulation

Definition (Triangulation). Eine Triangulation $\mathcal{T}$ einer Punktmenge $P = \{p_1, \dots, p_N\} \subset \mathbb{R}^2$ ist eine *maximale* „Familie“ von Dreiecken in der Form:

$$\mathcal{T} = \{T_i = \mathcal{C}(p_{i_1}, p_{i_2}, p_{i_3}) : \text{die Kanten } \overline{p_{i_1}p_{i_2}}, \overline{p_{i_2}p_{i_3}}, \overline{p_{i_1}p_{i_3}} \text{ schneiden sich mit keiner Kante } \overline{p_{k_1}p_{k_2}}, \overline{p_{k_2}p_{k_3}}, \overline{p_{k_1}p_{k_3}} \text{ eines anderen Dreiecks } T_k, ; k \neq i; i, k = 1, \dots, M_T\}$$

„*Maximal*“ bedeutet in diesem Kontext, dass kein weiteres Dreieck hinzugefügt werden kann, ohne dass sich Dreieckskanten schneiden.



Triangulation

Definition (Triangulation). Eine Triangulation $\mathcal{T}$ einer Punktmenge $P = \{p_1, \dots, p_N\} \subset \mathbb{R}^2$ ist eine *maximale* „Familie“ von Dreiecken in der Form:

$$\mathcal{T} = \{T_i = \mathcal{C}(p_{i_1}, p_{i_2}, p_{i_3}) \text{: die Kanten } \overline{p_{i_1}p_{i_2}}, \overline{p_{i_2}p_{i_3}}, \overline{p_{i_1}p_{i_3}} \text{ schneiden sich mit keiner Kante } \overline{p_{k_1}p_{k_2}}, \overline{p_{k_2}p_{k_3}}, \overline{p_{k_1}p_{k_3}} \text{ eines anderen Dreiecks } T_k, ; k \neq i; i, k = 1, \dots, M_T\}$$

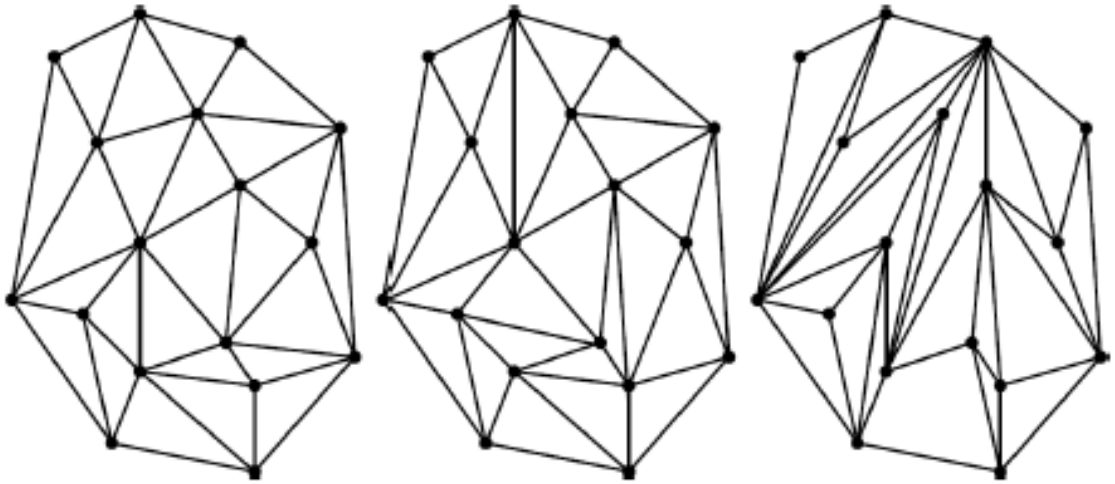
„*Maximal*“ bedeutet in diesem Kontext, dass kein weiteres Dreieck hinzugefügt werden kann, ohne dass sich Dreieckskanten schneiden.

Allgemeine Eigenschaften von $\mathcal{T}$ (Notation gemäß Online-Skript):

- Jedes $T \in \mathcal{T}$ mit $T = \mathcal{C}(p_i, p_j, p_k)$ ist **nicht-degeneriert**: $p_i \neq p_j \neq p_k$ und $\text{area}(T) > 0$
- Ein Dreieck $T = \mathcal{C}(p_i, p_j, p_k)$ **keinen anderen** Punkt $p_l \in \text{P}$ mit $l \neq i, j, k$
- $\mathcal{T}$ überdeckt die **komplette konvexe Hülle** von P: $\cup\, \mathcal{T} = [P]$
- Dreiecke überlappen sich, wenn überhaupt, nur an ihren Grenzen: $T_i \cap T_j = \emptyset \qquad \forall i \neq \text{j}$

Triangulation

Die Triangulation einer gegeben Punktmenge ist **NIEMALS** eindeutig. Es gibt immer mehrere Lösungen, um die gegebenen Punkt mit Dreiecken zu verbinden, welche die genannten Eigenschaften aufweisen.



Triangulation

Die Triangulation einer gegeben Punktmenge ist **NIEMALS** eindeutig. Es gibt immer mehrere Lösungen, um die gegebenen Punkt mit Dreiecken zu verbinden, welche die genannten Eigenschaften aufweisen.

Es lassen sich aber zumindest eine generelle Aussagen über die Triangulierung einer gebenen Punktmenge treffen:

Gegeben sei die Punktmenge $P = \{p_1, \dots, p_N\} \subset \mathbb{R}^2$ **(**in genereller Position). $\mathcal{T}(P)$ enthält immer

$$M_T = 2N - K - 2$$

Dreiecke und

$$M_K = 3N - K - 3$$

Kanten. K ist dabei die Anzahl der Punkte auf der konvexen Hülle von P.

Beweis: Euler-Poincaré-Theorem

⇒ **Die Anzahl der Dreiecke und Dreieckskanten ist konstant für alle möglichen $\mathcal{T}(P)$.**

Delaunay Triangulation

Sei $\mathcal{D}(\mathcal{V})$ der duale Graph bezüglich einer Voronoi Vermaschung $\mathcal{V}$.

Definition (Delaunay Triangulation). Eine Triangulation $\mathcal{T}$ wird als Delaunay Triangulation bezeichnet wenn

$$\mathcal{T} = \{T_i = \mathcal{C}(p_{i_1}, p_{i_2}, p_{i_3}) \colon \overline{p_{i_1}p_{i_2}}, \overline{p_{i_2}p_{i_3}}, \overline{p_{i_1}p_{i_3}} \in \mathcal{D}(\mathcal{V}), i = 1, \dots, M_T\}$$

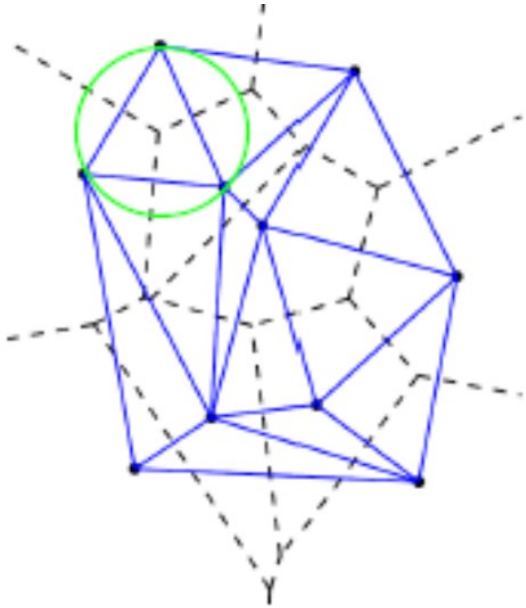
Delaunay Triangulation

Sei $\mathcal{D}(\mathcal{V})$ der duale Graph bezüglich einer Voronoi Vermaschung $\mathcal{V}$.

Definition (Delaunay Triangulation). Eine Triangulation $\mathcal{T}$ wird als Delaunay Triangulation bezeichnet wenn

$$\mathcal{T} = \{T_i = \mathcal{C}(p_{i_1}, p_{i_2}, p_{i_3}) \colon \overline{p_{i_1}p_{i_2}}, \overline{p_{i_2}p_{i_3}}, \overline{p_{i_1}p_{i_3}} \in \mathcal{D}(\mathcal{V}), i = 1, \dots, M_T\}$$

Lemma. Eine Triangulation $\mathcal{T}$ einer Punktmenge $P \in \mathbb{R}^2$ ist genau dann eine **Delaunay Triangulation**, wenn der Umkreis jedes Dreiecks $T \in \mathcal{T}$ keinen Punkt $p \in P$ in seinem Inneren enthält.



Konstruktive Delaunay Triangulation

Definition (Winkel-maximale Triangulation) Sei $\mathcal{T}$ eine Triangulation einer Punktmenge $P \subset \mathbb{R}^2$ und sei $A(\mathcal{T}) = (\alpha_1, \alpha_2, \dots, \alpha_{3M_T})$ die aufsteigend sortierte Sequenz der Dreieckswinkel. Des weiteren sei $A(\mathcal{T}') = (\alpha'_1, \alpha'_2, \dots, \alpha'_{3M_T})$ die aufsteigend sortierte Sequenz der Dreieckswinkel einer zweiten Triangulation $\mathcal{T}'$ von P.
Dann kann man sagen

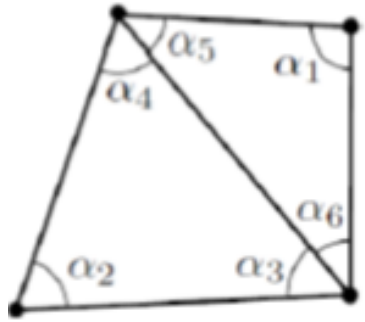
$$A(\mathcal{T}) > A(\mathcal{T}'),$$

falls ein Index I existiert, für den gilt

$$\alpha_i = \alpha'_i \text{ für } i < I \text{ und } \alpha_I > \alpha'_I.$$

Die Triangulation $\mathcal{T}$ wird als „Winkel-maximal“ bezeichnet, wenn für jede mögliche Triangulation $\mathcal{T}'(P)$ gilt:

$$A(\mathcal{T}) \geq A(\mathcal{T}')$$

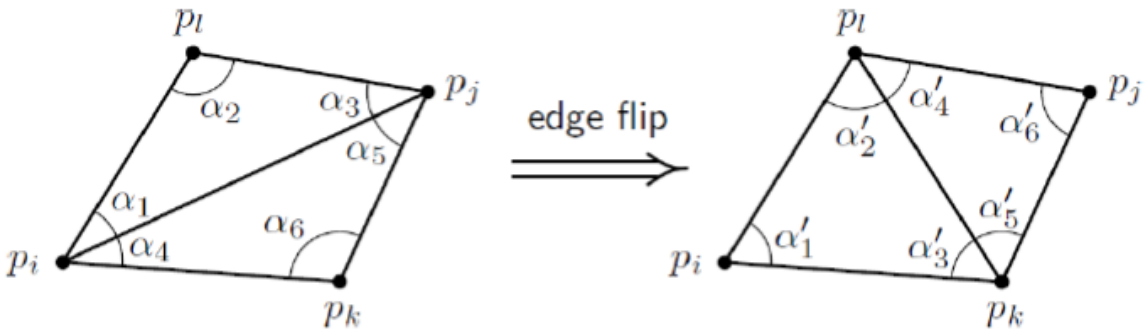


Konstruktive Delaunay Triangulation

Definition („illegale“ Kante, Kanten-flip) Sei $\mathcal{T}$ eine Triangulation einer Punktmenge $P \subset \mathbb{R}^2$ und seien $T_1 = \mathcal{C}(p_i, p_j, p_k)$, $T_2 = \mathcal{C}(p_i, p_j, p_l)$ zwei benachbarte Dreiecke dieser Triangulation, deren Vereinigung ein konvexes Viereck bilden.
Jetzt lässt sich ein weiteres Paar benachbarter Dreiecke $T'_1 = \mathcal{C}(p_i, p_l, p_k)$, $T'_2 = \mathcal{C}(p_j, p_k, p_l)$ definieren.
Wen für die beteiligten Winkel gilt:

$$\min_{1=1,\dots,6} \alpha_i < \min_{1=1,\dots,6} \alpha'_i$$

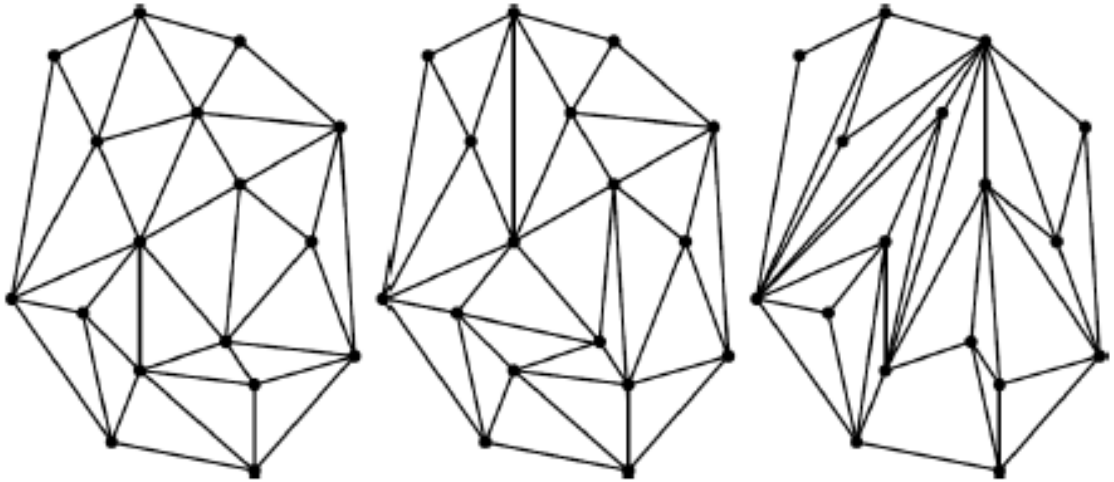
dann wird die Kante $\overline{p_i p_j}$ als **„illegale“ Kante** bezeichnet. Die Überführung von T_1, T_2 nach T'_1, T'_2 wird als „Kanten-flip“ (*edge flip*) bezeichnet.



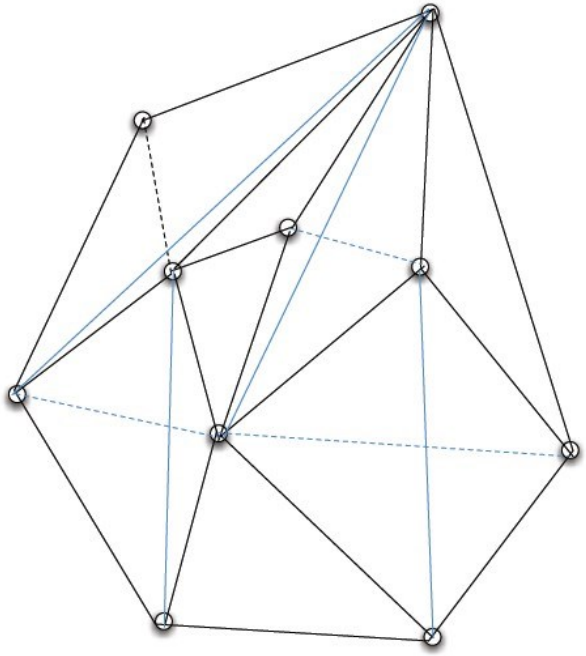
Konstruktive Delaunay Triangulation

Wenn eine Triangulation $\mathcal{T}$ eine „illegale“ Kante beinhaltet, führt ein Kanten-Flip zu einer neuen Triangulation $\mathcal{T}'$ mit $A(\mathcal{T}') > A(\mathcal{T})$. Wenn dies iterativ durchgeführt wird, bis keine „illegale“ Kanten mehr vorhanden sind, führt dies zu einer **Delaunay Triangulation**.

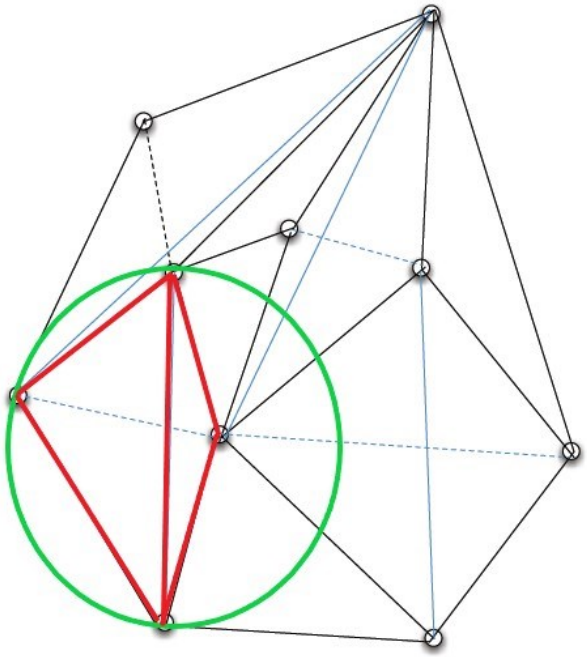
Theorem (Charakterisierung einer Delaunay Triangulation) Sei $\mathcal{T}$ eine Triangulation einer Punktmenge $P \subset \mathbb{R}^2$. $\mathcal{T}$ ist genau dann eine Delaunay Triangulation (im Sinne der Definition über die duale Vermaschung zu einer Voronoi Vermaschung), wenn sie keine „illegalen“ Kanten enthält.



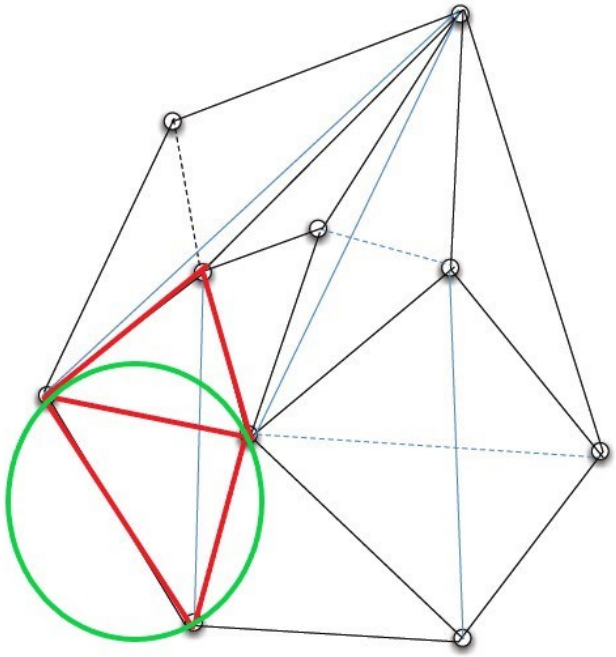
Konstruktive Delaunay Triangulation



Konstruktive Delaunay Triangulation



Konstruktive Delaunay Triangulation



Constrained Delaunay Triangulation

In bestimmten Situationen liegen bereits vor der Triangulierung wichtige Zusatzinformationen bezüglich der Nachbarschaftsbeziehungen zwischen Knoten vor.

Z.b.

- Bekannte Höhenlinien (Isolinien der Höhe, Linie mit konstanten Höhenwerten); diese Linienobjekte sollten als Kanten in einer Triangulierung erhalten bleiben.
- Linien, welche Störungen / Diskontinuitäten repräsentieren (diese Linien müssen erhalten bleiben, keine Dreieckskante darf ein solche Linie schneiden)

Constrained Delaunay Triangulation

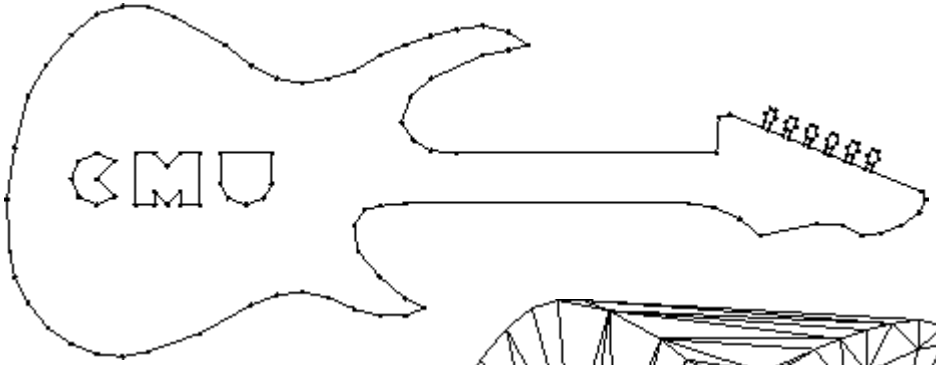
In bestimmten Situationen liegen bereits vor der Triangulierung wichtige Zusatzinformationen bezüglich der Nachbarschaftsbeziehungen zwischen Knoten vor.

Z.b.

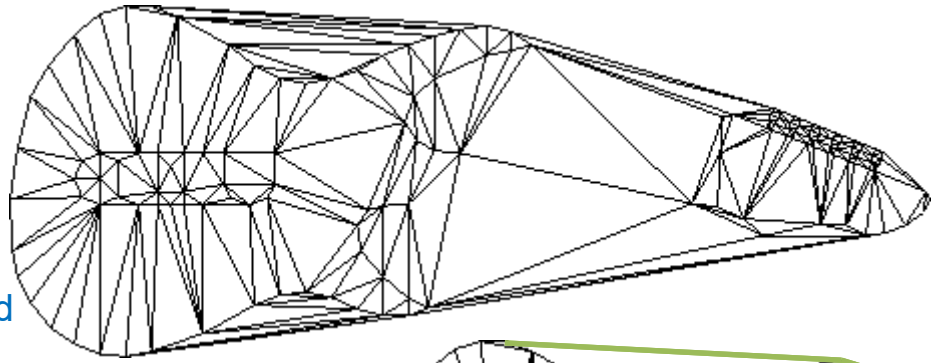
- Bekannte Höhenlinien (Isolinien der Höhe, Linie mit konstanten Höhenwerten); diese Linienobjekte sollten als Kanten in einer Triangulierung erhalten bleiben.
- Linien, welche Störungen / Diskontinuitäten repräsentieren (diese Linien müssen erhalten bleiben, keine Dreieckskante darf ein solche Linie schneiden)

Das **lokale Vorgehen** bei der Konstruktion einer Delaunay Triangulierung erlaubt es, diese Zusatzinformationen bzgl. Kanten direkt zu berücksichtigen. Die entstehende Triangulierung ist immer noch größenteils „winkel-maximal“, außer für die Dreiecke, welche durch die Zusatzinformationen beeinflusst wurden bzw. eine solche „constrained“ Kante verwenden.

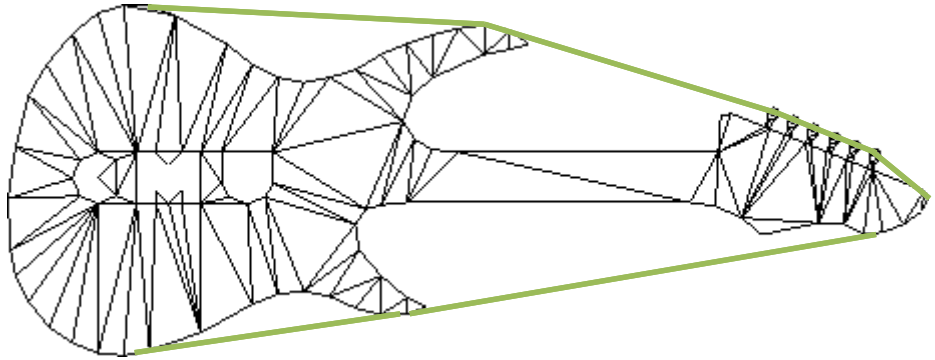
Constrained Delaunay Triangulation



Gegeben seinen die Umrisskanten / Knoten einer Gitarre.

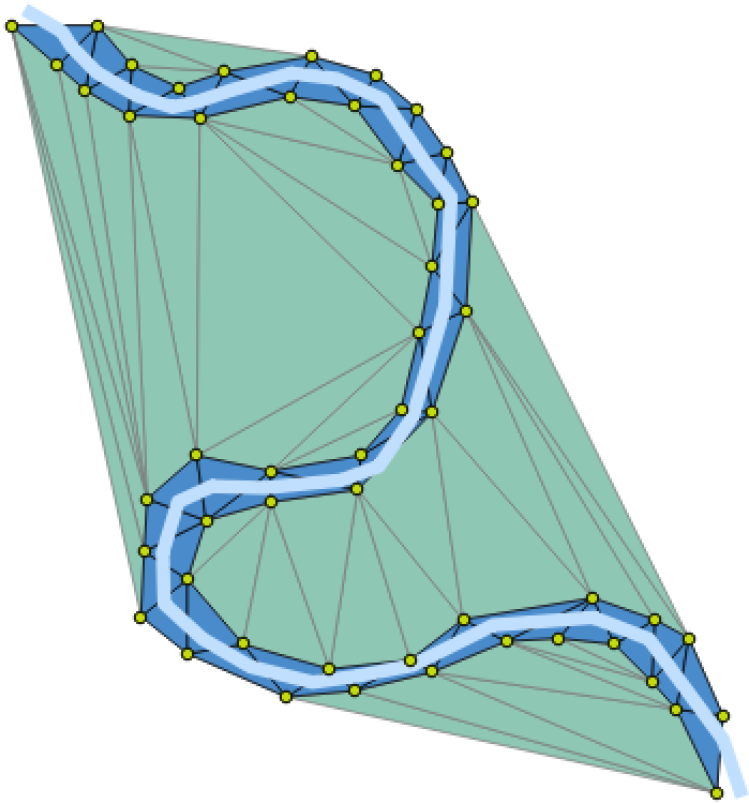


Eine Triangulierung ohne Constraints führt zu folgendem Ergebnis. Einige Details der Gitarre gehen verloren und es wird die komplette konvexe Hülle vermascht.



Werden die Umrisskanten als Constraints verwendet, bleiben alle Details erhalten. Es wird zwar immer noch die komplette konvexe Hüllte trianguliert, alle „äußeren“ Dreiecke lassen sich aber nachträglich entfernen.

Constrained Delaunay Triangulation



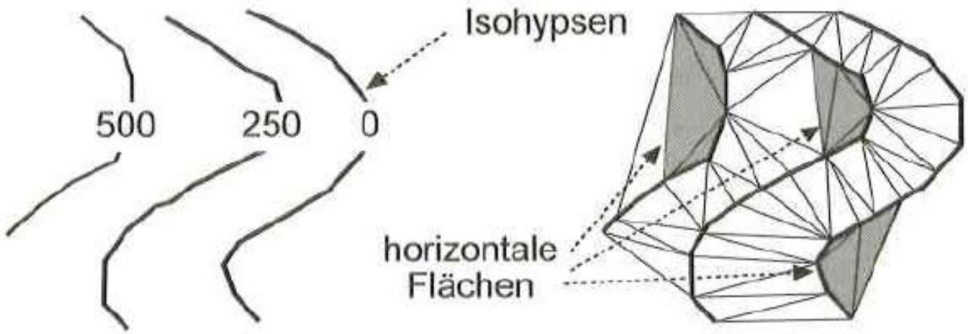
Triangulierung mittels Constraints welche das Flussbett beschreiben.



Triangulierung einer Insel mit der Küstenlinie als Constraints. Alle äußeren Dreiecke wurden nachträglich entfernt.

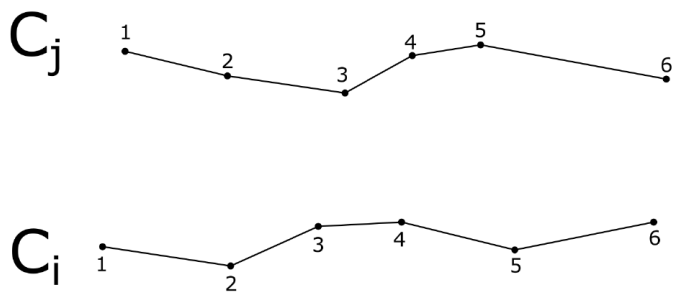
Parkettierung (*tiling*)

Triangulierung von Linienobjekten (z.B. Isohypsen, Isobaren) kann prinzipiell etwa als Constrained Delaunay Triangulation aufgefasst werden. Allerdings können dann u.U. ungewünschte Effekte auftreten (z.B. Plateaueffekte):



Parkettierung

Triangulierung von Linienobjekten (z.B. Isohypsen, Isobaren) kann prinzipiell etwa als Constrained Delaunay Triangulation aufgefasst werden. Allerdings können dann u.U. ungewünschte Effekte auftreten (z.B. Plateaueffekte):

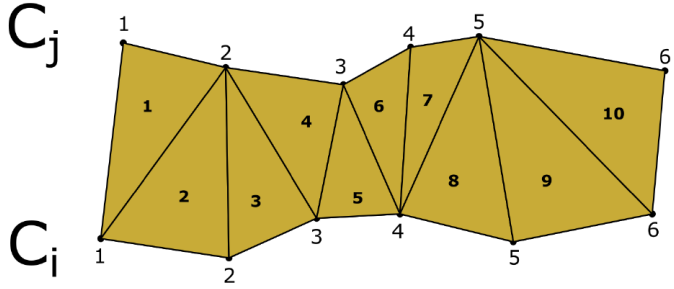


Es seien zwei Folgen $C_i = \{p_1, \ldots, p_n\}$ und $C_j = \{q_1, \ldots, q_m\}$ von Punkten gegeben.
Eine **Parkettierung (Tiling)** mittels Dreiecken ist dann gegeben, wenn folgende Bedingungen erfüllt sind:

1. aufeinanderfolgende Punkte eines Linienobjektes C_i oder C_j sind durch eine Dreieckskante verbunden,
2. jedes Dreieck enthält maximal zwei Punkte, die in demselben Linienobjekt C_i oder C_j enthalten sind.

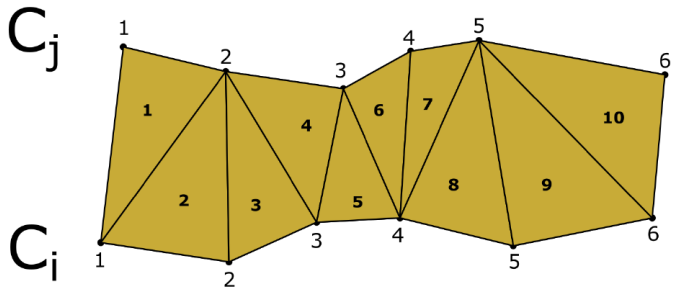
Parkettierung

Eine Parkettierung muss nicht zwangsläufig die gesamte konvexe Hülle aller Punkte umfassen. Ebenso ist es nicht ausgeschlossen, dass sich 3D Dreiecke überschneiden, wenn sie in die 2D Ebene projiziert werden (eine Parkettierung stellt also keine klassische Triangulierung dar).

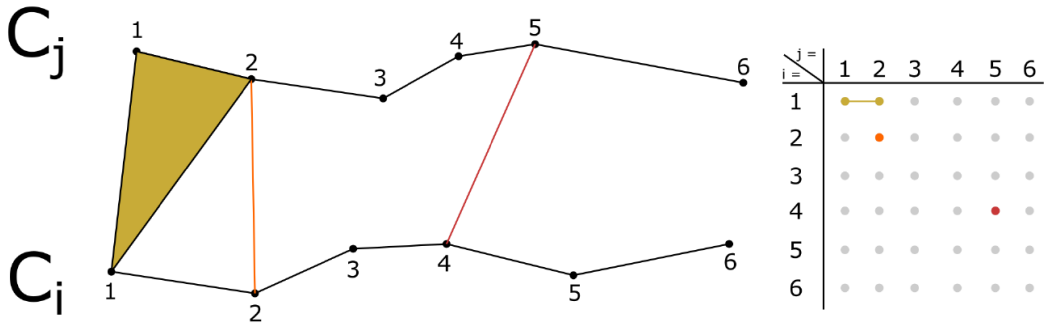


Parkettierung

Eine Parkettierung muss nicht zwangsläufig die gesamte konvexe Hülle aller Punkte umfassen. Ebenso ist es nicht ausgeschlossen, dass sich 3D Dreiecke überschneiden, wenn sie in die 2D Ebene projiziert werden (eine Parkettierung stellt also keine klassische Triangulierung dar).

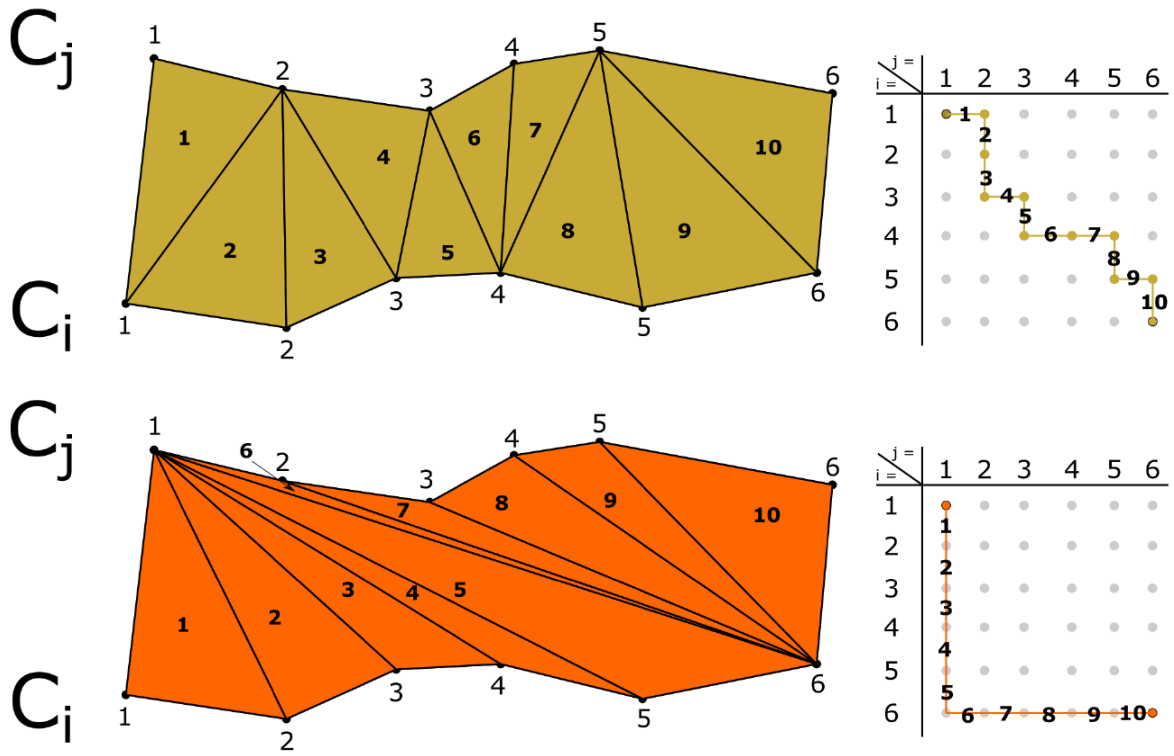


Parkettierungen können mit Hilfe eines Suchgraphens dargestellt werden: jeder Eintrag der Matrix beschreibt eine Verbindungslinie zwischen den Punkten der jeweiligen Zeile und Spalte.



Parkettierung

Jede Parkettierung kann mittels eines Pfades in dem Suchgraphen dargestellte werden



Parkettierung

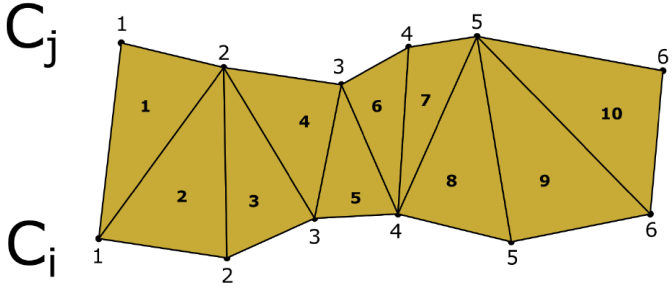
Eine **Gewichtungen der Kanten** in dem Suchgraphen erlauben es, die in einem gewissen Sinne **optimale Parkettierungen** zu finden (also Parkettierungen, die eine gewissen Zielfunktion minimieren; etwa die Gesamtfläche der Parkettierung).

Dazu stellt man Graphen mit gewichteten Kanten auf: einzeln gewichtet, als auch kumulativ gewichtet.

Parkettierung

Eine **Gewichtungen der Kanten** in dem Suchgraphen erlauben es, die in einem gewissen Sinne **optimale Parkettierungen** zu finden (also Parkettierungen, die eine gewissen Zielfunktion minimi rung).

Dazu stellt man G zeln gewichtet, als auch kumulativ gewichtet.



Gewichte

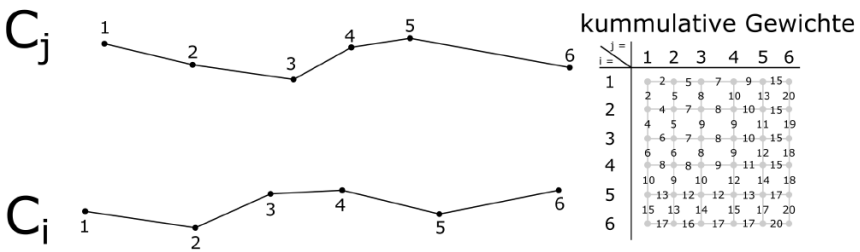
j = / i =	1	2	3	4	5	6
1	2	3	2	2	6	
2	2	3	3	3	4	5
3	2	3	1	2	5	
4	2	1	2	1	1	4
5	2	2	1	2	5	
6	2	1	1	1	2	3
	2	2	1	2	4	
	4	3	2	3	3	3
	3	3	2	1	4	
	5	4	4	3	4	3
	2	3	3	2	3	

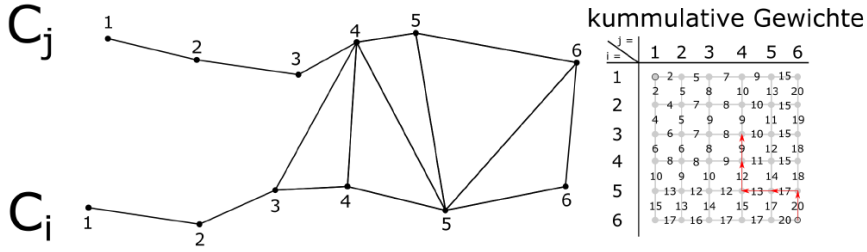
kummulative Gewichte

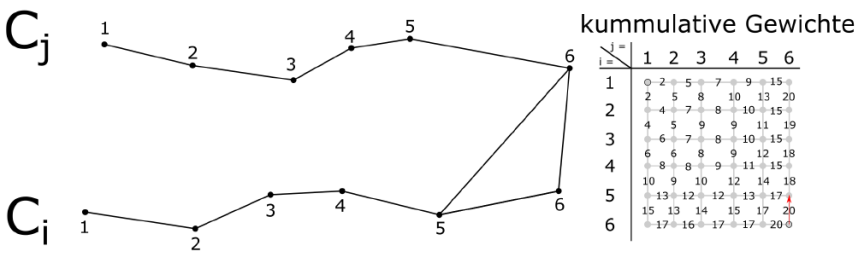
j = / i =	1	2	3	4	5	6
1	2	5	7	9	15	
2	2	5	8	10	13	20
3	4	7	8	10	15	
4	4	5	9	9	11	19
5	6	7	8	10	15	
6	6	6	8	9	12	18
	8	8	9	11	15	
	10	9	10	12	14	18
	13	12	12	13	17	
	15	13	14	15	17	20
	17	16	17	17	20	

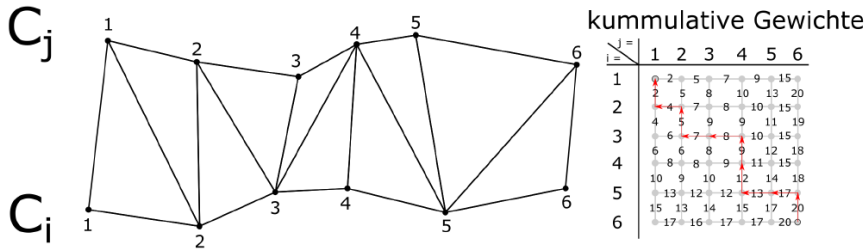
Parkettierung

Die optimale Parkettierung ergibt sich durch den kürzesten Pfad im kumuliert gewichteten Graphen. Diese optimale Parketierung muss nicht eindeutig sein.





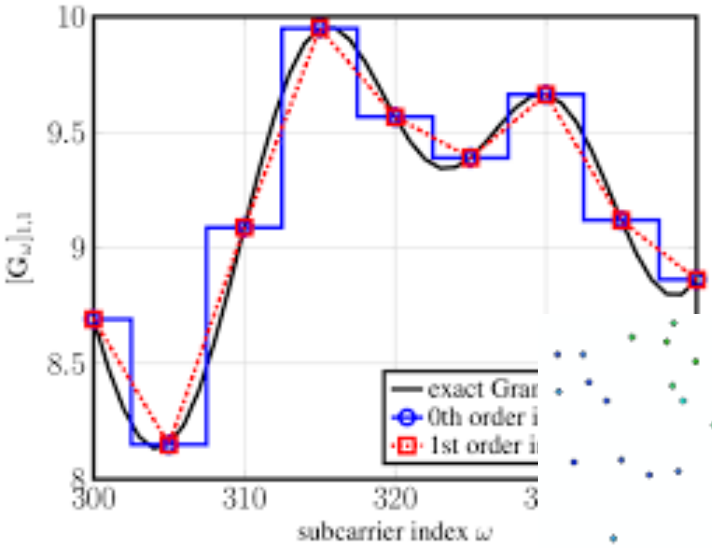


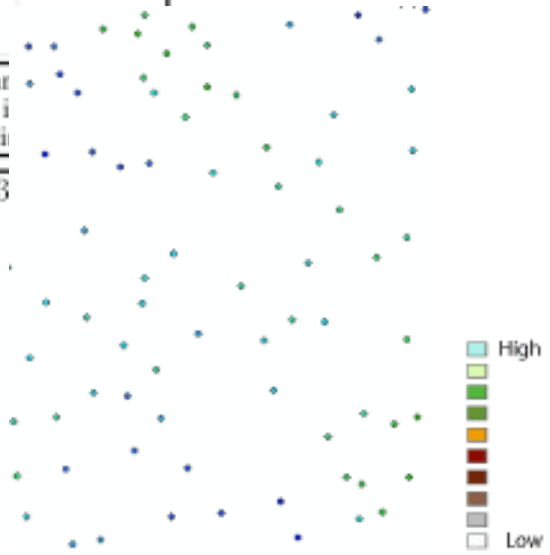


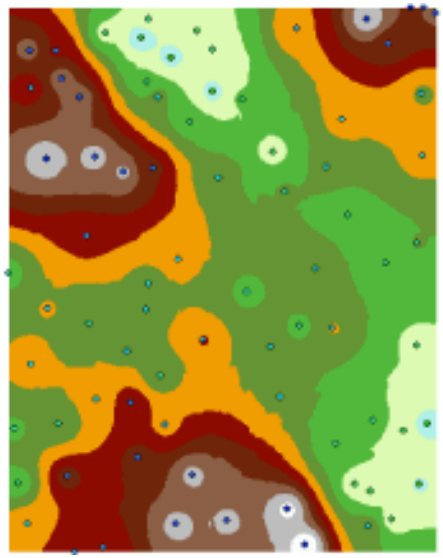
Übersicht

1. Definitionen, Funktionen, Anwendungen
2. Koordinatensysteme und -transformationen
3. Räumliche Datenmodellierung
4. Vermaschungen
5. **Räumliche Interpolation**
6. Transformationen, Filtermethoden

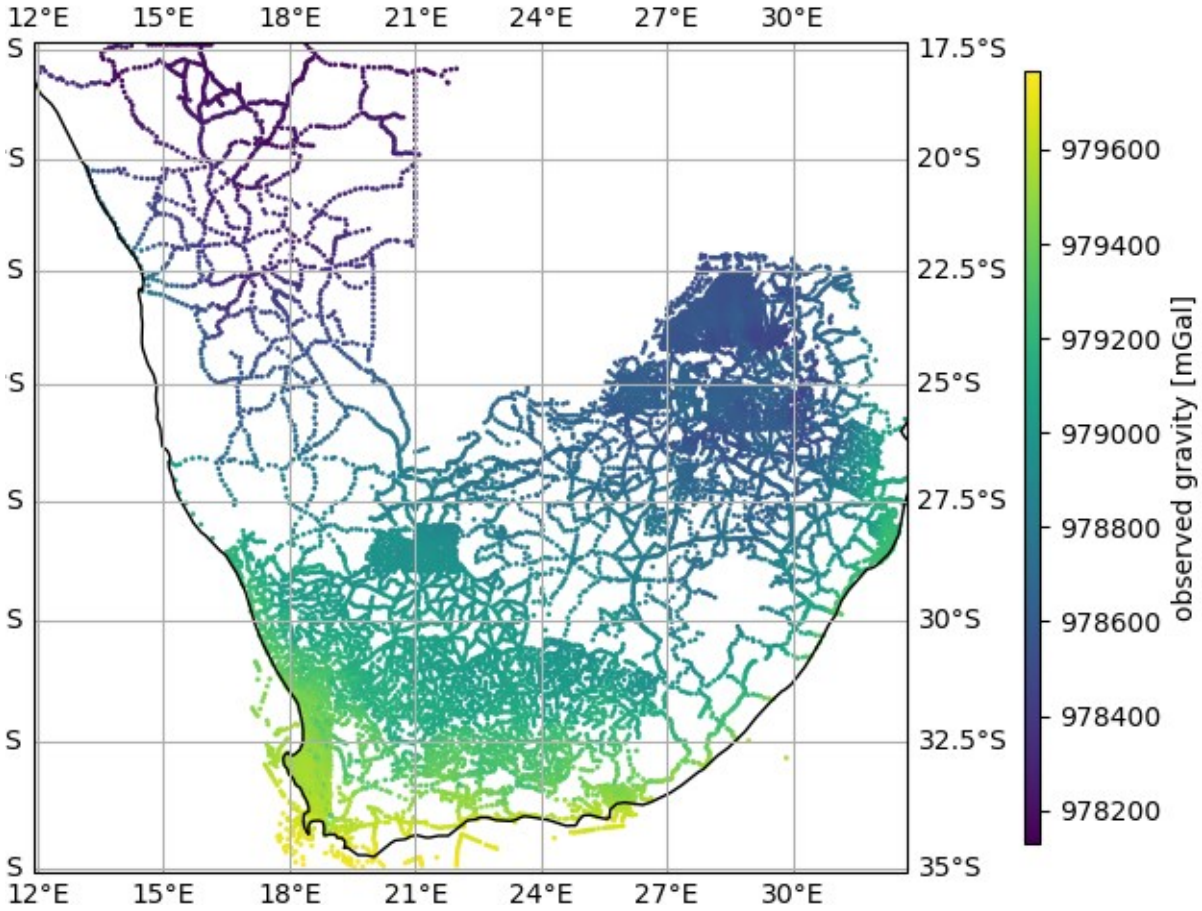
Beispiele


10
9.5
9
8.5
8
300
310
320
subcarrier index ω
exact Gra
0th order i
1st order i


High
Low

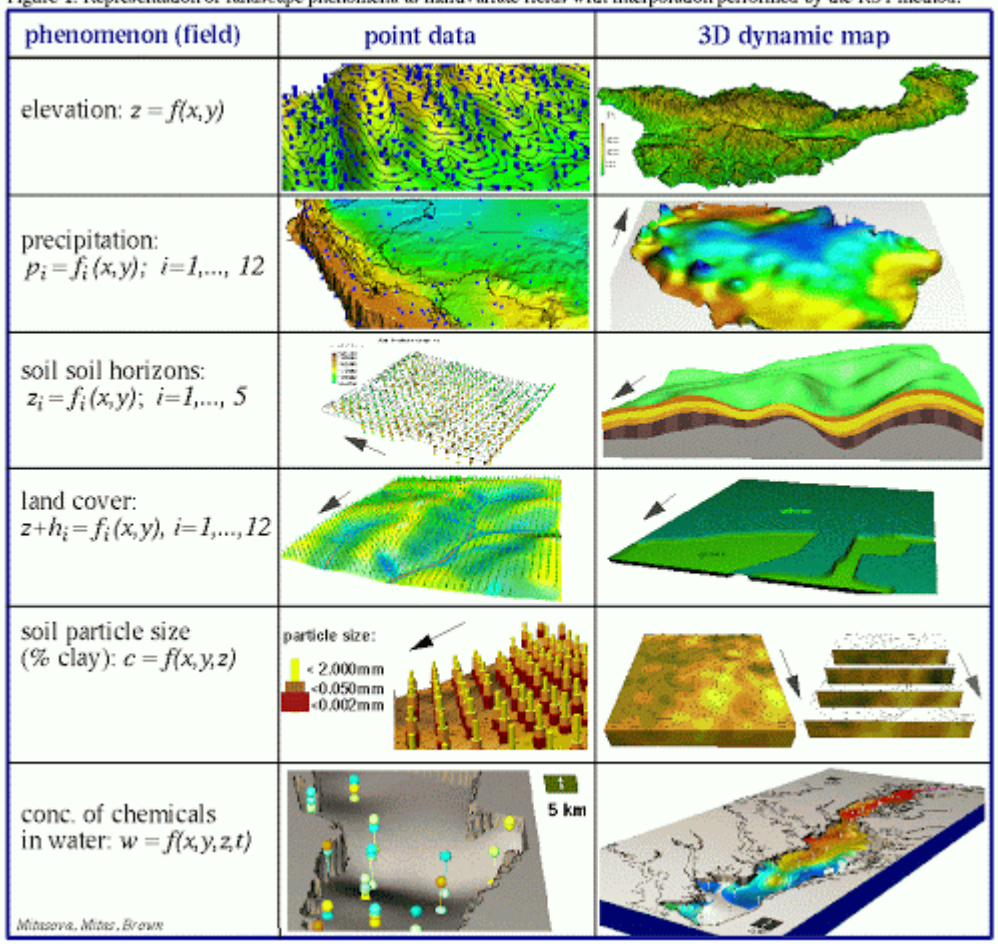


Beispiele


12°E
15°E
18°E
21°E
24°E
27°E
30°E
17.5°S
20°S
22.5°S
25°S
27.5°S
30°S
32.5°S
35°S
979600
979400
979200
979000
978800
978600
978400
978200
observed gravity [mGal]

Beispiele

Figure 1. Representation of landscape phenomena as multivariate fields with interpolation performed by the RST method.



fatra.cnr.ncsu.edu/~hmitaso/gmslab/asae97/mitasova973034.html

Interpolation

Im Allgemeinen liegen Messwerte nur an endlich vielen Positionen $x_1, \dots, x_N$ vor (dies können physische Positionen im Raum, Zeitpunkte oder andere abstrakte Positionen sein). Für jede Position x_i liegt ein Messwert y_i für einen Parameter vor.

Das Ziel einer Interpolation ist die „Vorhersage“ des gesuchten Parameters y an einer beliebigen Position x, an dem initial kein Messwert vorliegt.

Dies erfolgt zumeist, indem eine Funktion f, mit

$f(x_i) = y_i, i = 1, \dots, N$ (**klassische / exakte Interpolation**) oder zumindest

$f(x_i) \approx y_i, i = 1, \dots, N$ (**Approximation**),

gefunden wird, welche bestimmte Eigenschaften aufweisen soll.

Die „Vorhersage“ für eine Position x erfolgt dann über $y = f(x)$.

Interpolation

Gegeben sei $(x_i, y_i), i = 1, \ldots, N$. Die Interpolationsfunktion liegt zumeist in einen von zwei Formen vor, entweder

$$f(x) = \sum_{i=1}^{N} \lambda_i(x) y_i \,,$$

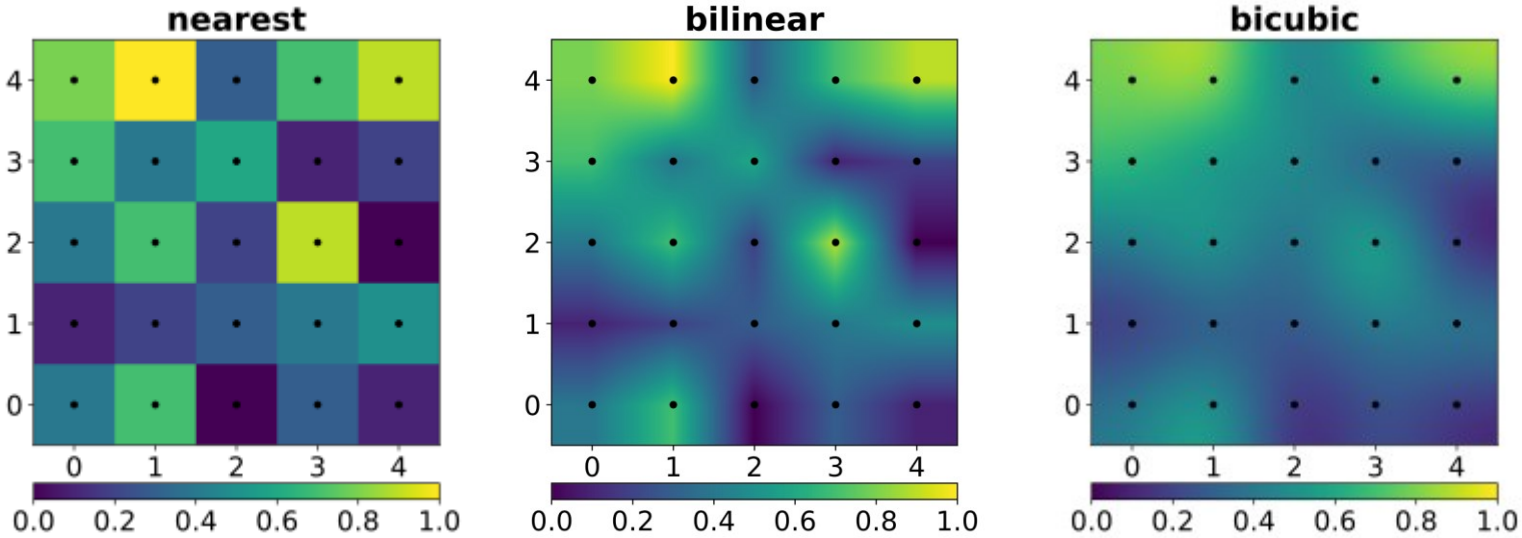
wobei hier die Gewichte λ_i von x abhängen, oder

$$f(x) = \sum_{i=1}^{N} \lambda_i f_i(x) \,,$$

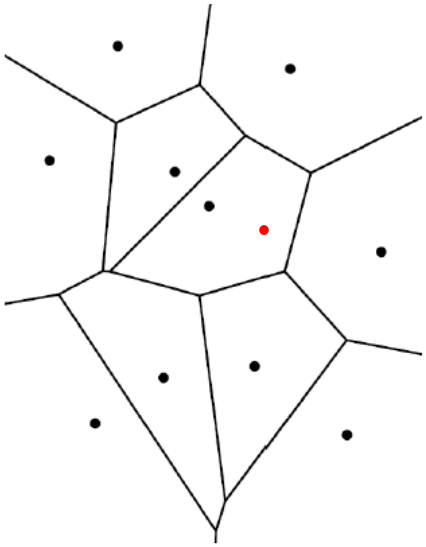
mit konstanten Gewichten λ_i(abhängig von y_i).

Gitter-basierte / Element-basierte Interpolation

- **Nearets Neighbour Interpolation** (basiert auf Voronoi Vermaschung): $f(x)$ diskontinuierlich
- **Natural Neighbour Interpolation** (Voronoi Vermaschung benötigt): $f(x)$ kontinuierlich + differenzierbar
- **Lineare Interpolation auf Triangulationen**: $f(x)$ kontinuierlich
- **Bilinear Interpolation** (benötigt Rechteck-Gitter): $f(x)$ kontinuierlich



Nearest Neighbour Interpolation



Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ die Voronoi Vermaschung der Punktmenge $P = \{p_1, \dots, p_N\}$ und sei y_i ein Parameterwert an einem Punkt p_i, dann

$$f(x) = \sum_{i=1}^{N} \lambda_i(x) y_i$$

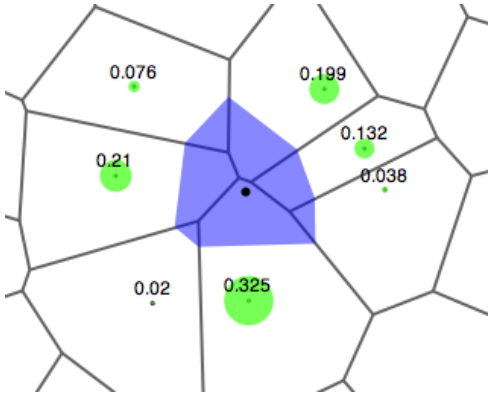
mit den Gewichten

$$\lambda_i(x) = \begin{cases} 1, x \in V_i \\ 0, x \notin V_i \end{cases}$$

Eigenschaften:

- $f(x)$ diskontinuierlich;
- Lokaler Interpolator, der nur auf (p_i, y_i) basiert, wenn $x \in V_i$;
- Extrapolation ist konzeptuell möglich.

Natural Neighbour Interpolation



Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ die Voronoi Vermaschung der Punktmenge $P = \{p_1, \dots, p_N\} \subset \mathbb{R}^n$ und sei y_i ein Parameterwert an einem Punkt p_i. Sei des weiteren $x \in \mathbb{R}^n$ ein beliebiger Punkt und $\mathcal{V}'$ die Voronoi Vermaschung bezüglich $P \cup \{x\}$. Sind die Punkte $p_{i_x}, \mathrm{i} = 1, \dots, n_x$ die **starken** Voronoi Nachbarn von x bezüglich $\mathcal{V}'$, dann gilt

$$\lambda_{i_x} = \frac{\mathrm{area}(V'(x) \cap V(p_{i_x}))}{\mathrm{area}(V'(x))}.$$

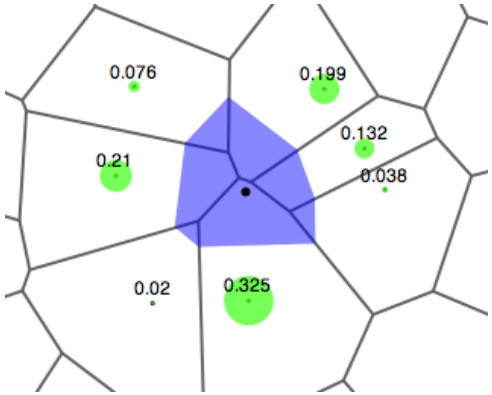
Sind Messwerte y_i für die Punkte p_i gegeben, kann man $f(x)$ wiefolgt vorhersagen

$$f(x) = \sum_{i=1}^{n_x} \lambda_{i_x} y_{i_x}.$$

Für $\lambda_{i_x} > 0$ gilt $\sum_{i=1}^{n_x} \lambda_{i_x} = 1$.

Dies ist eine typische Eigenschaft von Interpolationsgewichten. (Isoparametrischer Interpolator)

Natural Neighbour Interpolation



Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ die Voronoi Vermaschung der Punktmenge $P = \{p_1, \dots, p_N\} \subset \mathbb{R}^n$ und sei y_i ein Parameterwert an einem Punkt p_i. Sei des weiteren $x \in \mathbb{R}^n$ ein beliebiger Punkt und $\mathcal{V}'$ die Voronoi Vermaschung bezüglich $P \cup \{x\}$. Sind die Punkte $p_{i_x}, \mathrm{i} = 1, \dots, n_x$ die **starken** Voronoi Nachbarn von x bezüglich $\mathcal{V}'$, dann gilt

$$\lambda_{i_x} = \frac{\mathrm{area}(V'(x) \cap V(p_{i_x}))}{\mathrm{area}(V'(x))}.$$

Sind Messwerte y_i für die Punkte p_i gegeben, kann man $f(x)$ wiefolgt vorhersagen

$$f(x) = \sum_{i=1}^{n_x} \lambda_{i_x} y_{i_x}.$$

Eigenschaften:

- $f(x)$ ist kontinuierlich und differenzierbar;
- Lokaler Interpolator, der nur Tupel (p_i, y_i) berücksichtigt, die die starken Voronoi-Nachbarn von x sind;
- Extrapolation ist konzeptuell möglich.

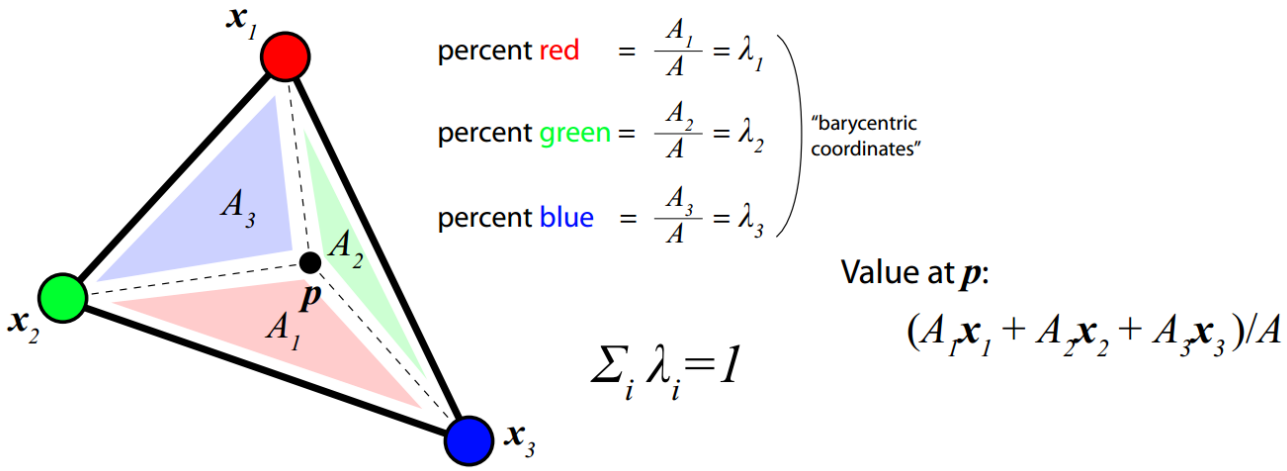
Lineare Interpolation auf Dreiecken

Gegeben seien die Datentupel $(x_i, y_i), (x_j, y_j), (x_k, y_k)$ und sei x ein Punkt innerhalb eines Dreiecks $\mathcal{C}(x_i, x_j, x_k)$. Punkt x lässt sich als Linearkombination

$$x = \lambda_i x_i + \lambda_j x_j + \lambda_k x_k$$

ausdrücken, gleiches gilt für den unbekannten Messwert am Punkt x:

$$f(x) = \lambda_i y_i + \lambda_j y_j + \lambda_k y_k$$



Lineare Interpolation auf Dreiecken

Gegeben seien die Datentupel $(x_i, y_i), (x_j, y_j), (x_k, y_k)$ und sei x ein Punkt innerhalb eines Dreiecks $\mathcal{C}(x_i, x_j, x_k)$. Punkt x lässt sich als Linearkombination

$$x = \lambda_i x_i + \lambda_j x_j + \lambda_k x_k$$

ausdrücken, gleiches gilt für den unbekannten Messwert am Punkt x:

$$f(x) = \lambda_i y_i + \lambda_j y_j + \lambda_k y_k$$

Die Koeffizienten / Gewichte werden als **baryzentrische Koodinaten** von x (in 2D). Seien die Koordinaten von $x = (x^{(1)}, x^{(2)})$, dann können die Koeffizienten wie folgt ausgedrückt werden

$$\lambda_i = \frac{\det(A_i)}{\det(A)}, \qquad \lambda_j = \frac{\det(A_j)}{\det(A)}, \qquad \lambda_k = \frac{\det(A_k)}{\det(A)}$$

mit den Matrizen

$$A = \begin{pmatrix} x_i^{(1)} & x_j^{(1)} & x_k^{(1)} \\ x_i^{(2)} & x_j^{(2)} & x_k^{(2)} \\ 1 & 1 & 1 \end{pmatrix}, A_i = \begin{pmatrix} x^{(1)} & x_j^{(1)} & x_k^{(1)} \\ x^{(2)} & x_j^{(2)} & x_k^{(2)} \\ 1 & 1 & 1 \end{pmatrix}, A_j = \begin{pmatrix} x_i^{(1)} & x^{(1)} & x_k^{(1)} \\ x_i^{(2)} & x^{(2)} & x_k^{(2)} \\ 1 & 1 & 1 \end{pmatrix}, A_k = \begin{pmatrix} x_i^{(1)} & x_j^{(1)} & x^{(1)} \\ x_i^{(2)} & x_j^{(2)} & x^{(2)} \\ 1 & 1 & 1 \end{pmatrix}$$

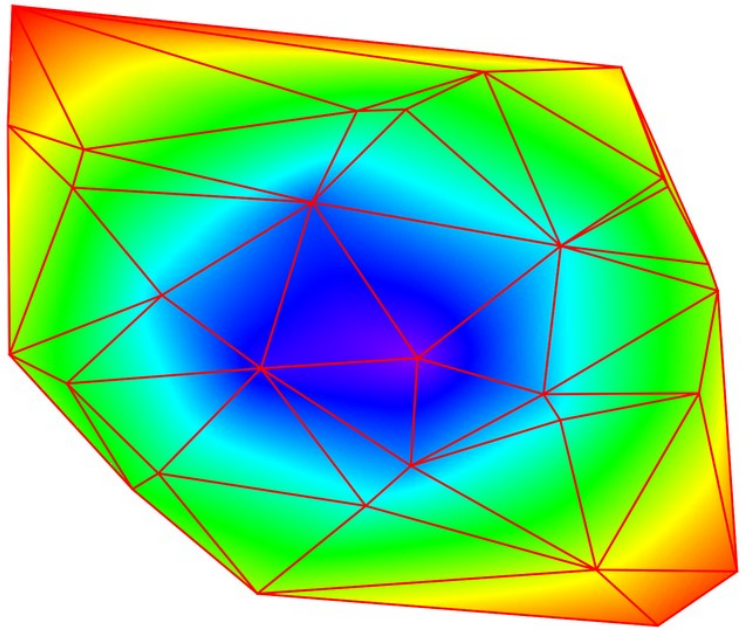
Lineare Interpolation auf Dreiecken

Gegeben seien die Datentupel $(x_i, y_i), (x_j, y_j), (x_k, y_k)$ und sei x ein Punkt innerhalb eines Dreiecks $\mathcal{C}(x_i, x_j, x_k)$. Punkt x lässt sich als Linearkombination

$$x = \lambda_i x_i + \lambda_j x_j + \lambda_k x_k$$

ausdrücken, gleiches gilt für den unbekannten Messwert am Punkt x:

$$f(x) = \lambda_i y_i + \lambda_j y_j + \lambda_k y_k$$



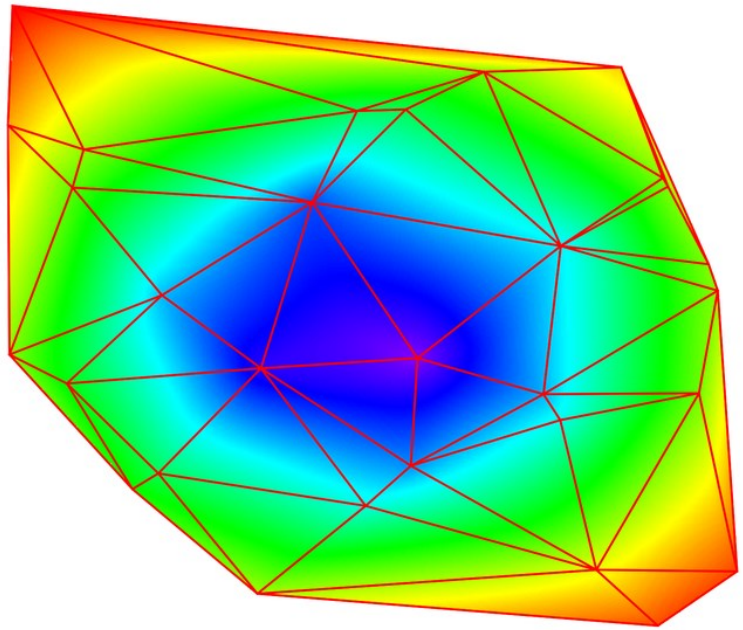
Lineare Interpolation auf Dreiecken

Gegeben seien die Datentupel $(x_i, y_i), (x_j, y_j), (x_k, y_k)$ und sei x ein Punkt innerhalb eines Dreiecks $\mathcal{C}(x_i, x_j, x_k)$. Punkt x lässt sich als Linearkombination

$$x = \lambda_i x_i + \lambda_j x_j + \lambda_k x_k$$

ausdrücken, gleiches gilt für den unbekannten Messwert am Punkt x:

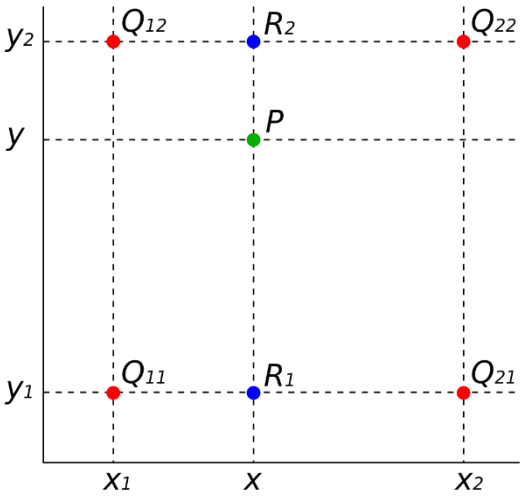
$$f(x) = \lambda_i y_i + \lambda_j y_j + \lambda_k y_k$$



Eigenschaften:

- $f(x)$ ist kontinuierlich
- Auf Dreieckskanten nicht stetig differenzierbar
- lokaler Interpolator der nur $(x_i y_i), (x_j, y_j), (x_k, y_k)$ berücksichtigt, wenn $x \in \mathcal{C}(x_i, x_j, x_k)$
- Extrapolation ist konzeptuell **NICHT** möglich

Bilineare Interpolation auf einem Punktgitter



Seien die Messwerte $y_{i,j}$ auf einem Punktgitter $Q_{i,j} \in \mathbb{R}^2$ gegeben. Der unbekannte Wert an einem Punkt $P = (x^{(1)}, y^{(2)})$ innerhalb eines Gitterelements $Q_{i,j}, Q_{i+1,j}, Q_{i,j+1}, Q_{i+1,j+1}$ kann wie folgt vorhergesagt werden:

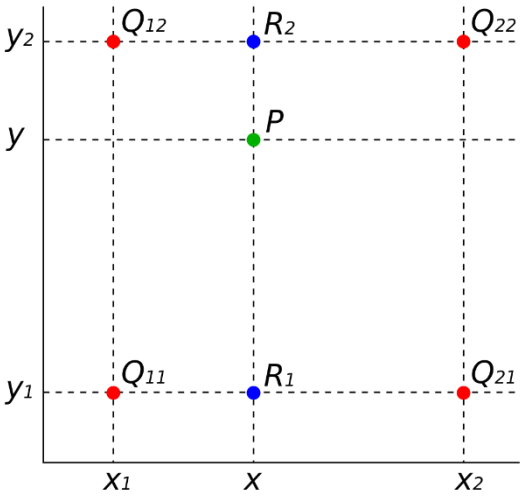
$$f\left(x^{(1)}, x^{(2)}\right) = \frac{x_{i,j+1}^{(2)} - x^{(2)}}{x_{i,j+1}^{(2)} - x_{i,j}^{(2)}} f_{i,j} + \frac{x^{(2)} - x_{i,j}^{(2)}}{x_{i,j+1}^{(2)} - x_{i,j}^{(2)}} f_{i,j+1}$$

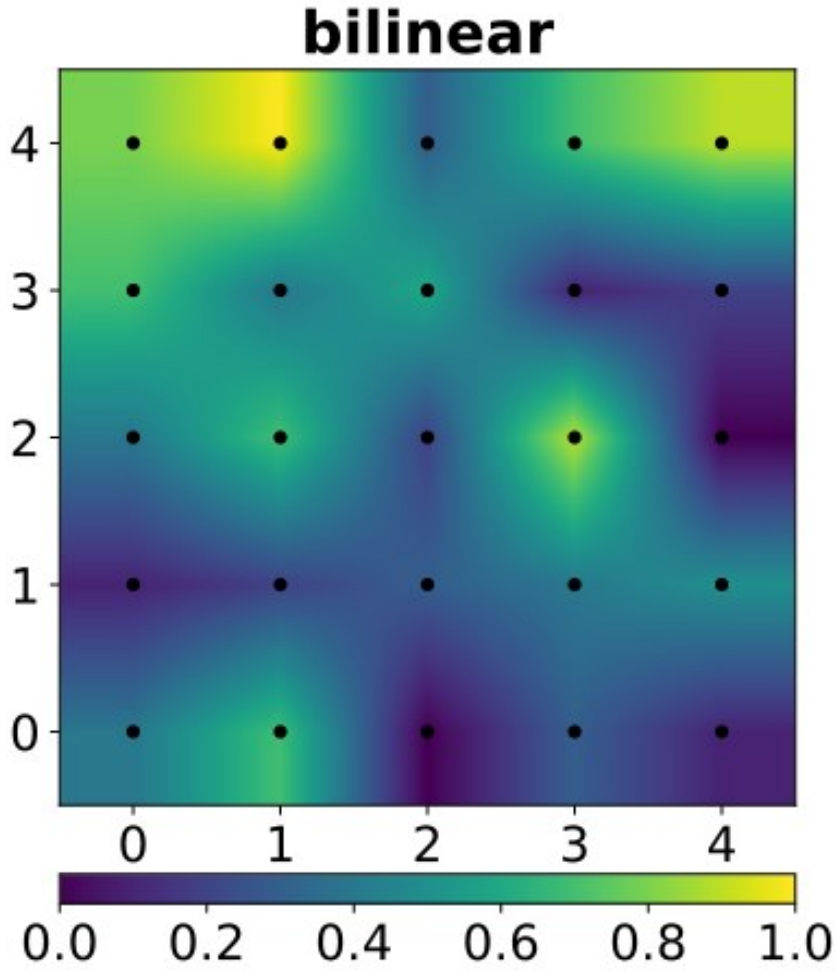
mit

$$f_{i,j} = \frac{x_{i+1,j}^{(1)} - x^{(1)}}{x_{i+1,j}^{(1)} - x_{i,j}^{(1)}} y_{i,j} + \frac{x^{(1)} - x_{i,j}^{(1)}}{x_{i+1,j}^{(1)} - x_{i,j}^{(1)}} y_{i+1,j}$$

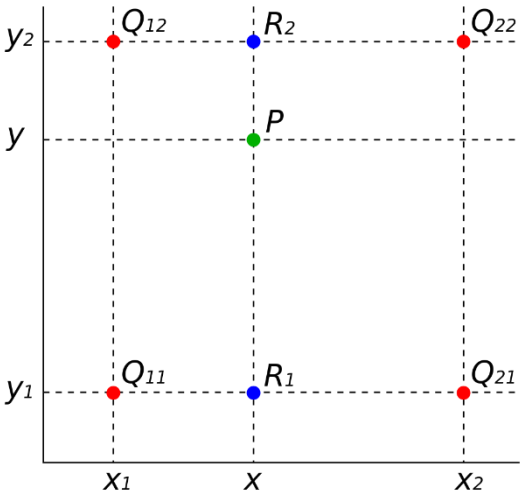
$$f_{i,j+1} = \frac{x_{i+1,j}^{(1)} - x^{(1)}}{x_{i+1,j}^{(1)} - x_{i,j}^{(1)}} y_{i,j+1} + \frac{x^{(1)} - x_{i,j}^{(1)}}{x_{i+1,j}^{(1)} - x_{i,j}^{(1)}} y_{i+1,j+1}$$

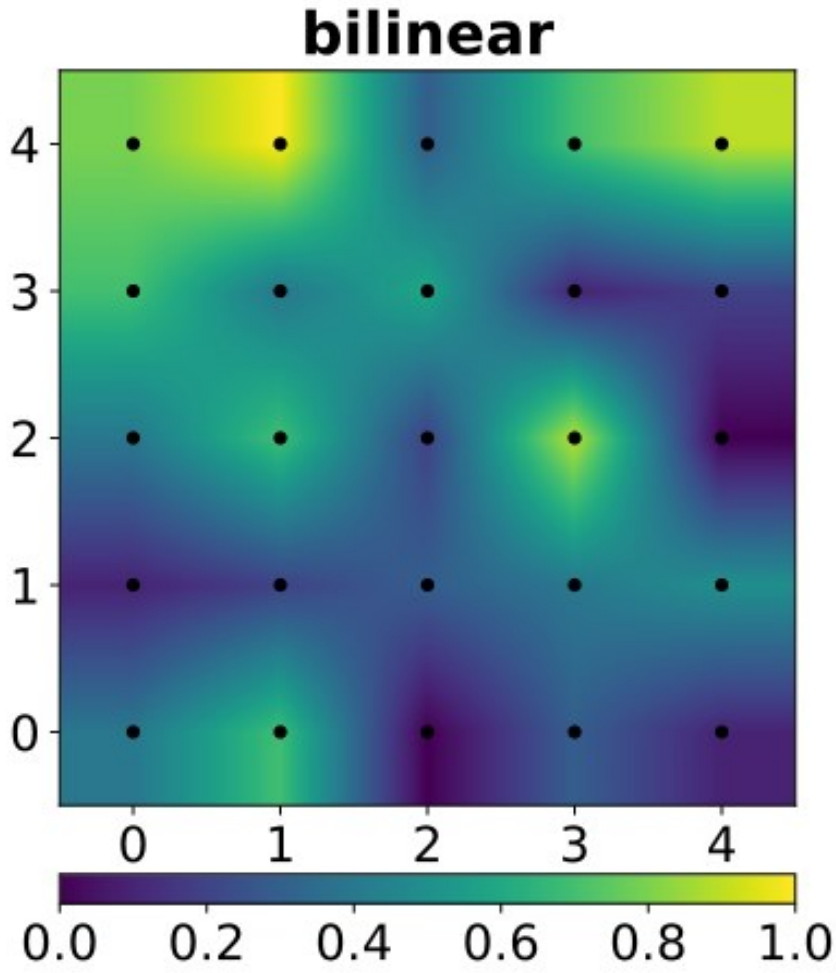
Bilinear Interpolation





Bilinear Interpolation

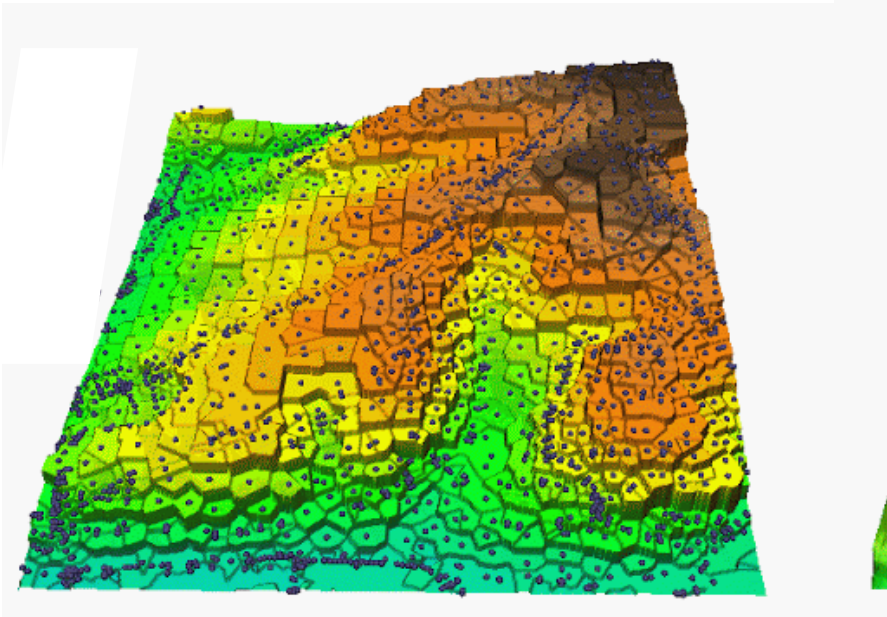


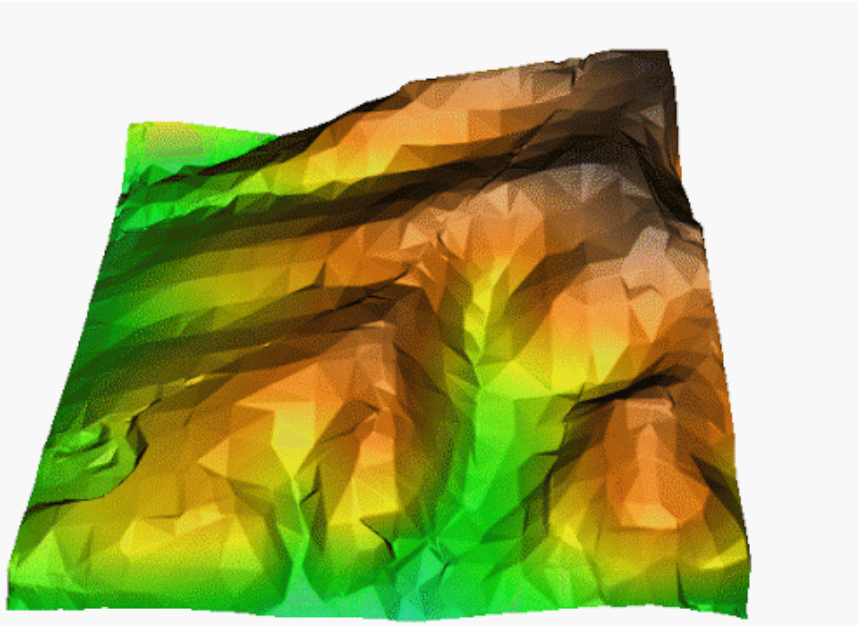


Eigenschaften:

- $f(x)$ ist kontinuierlich
- auf Gitterkanten nicht differenzierbar
- Lokaler Interpolator, der sich nur auf
  $(x_{i,j}y_{i,j}), (x_{i+1,j}, y_{i+1,j}),$
  $(x_{i,j+1}, y_{i,j+1}), (x_{i+1,j+1}, y_{i+1,j+1})$
  bezieht, wenn x innerhalb dieses Gitterelements ist
- Extrapolation ist konzeptionell nicht möglich

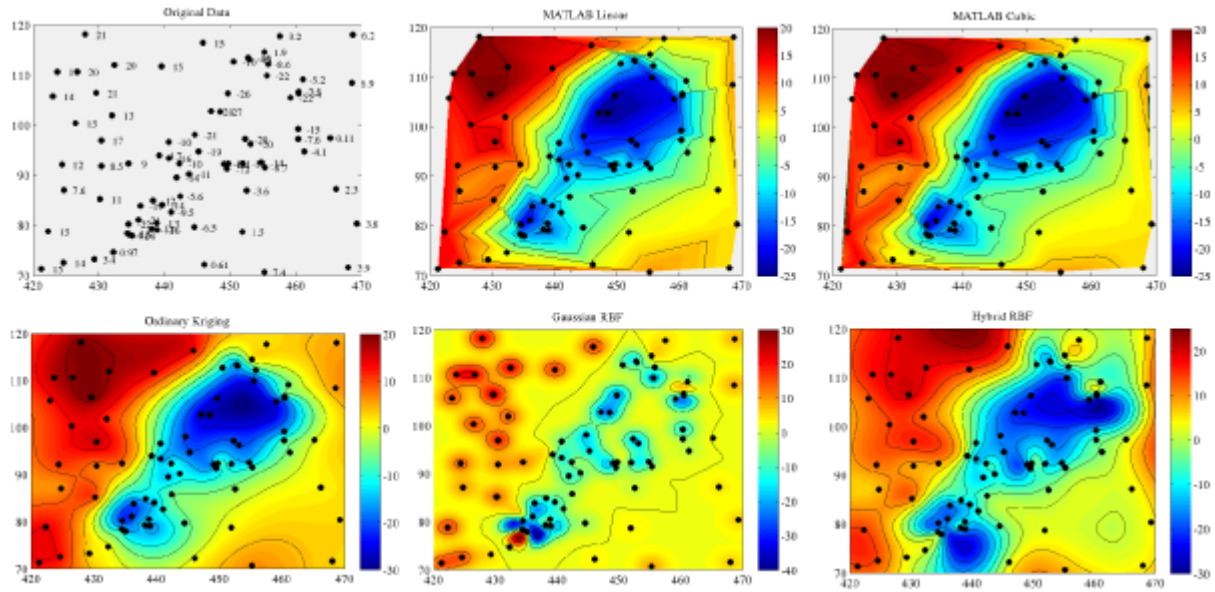
Nearest Neighbour vs Lin. Interp. auf Dreiecken





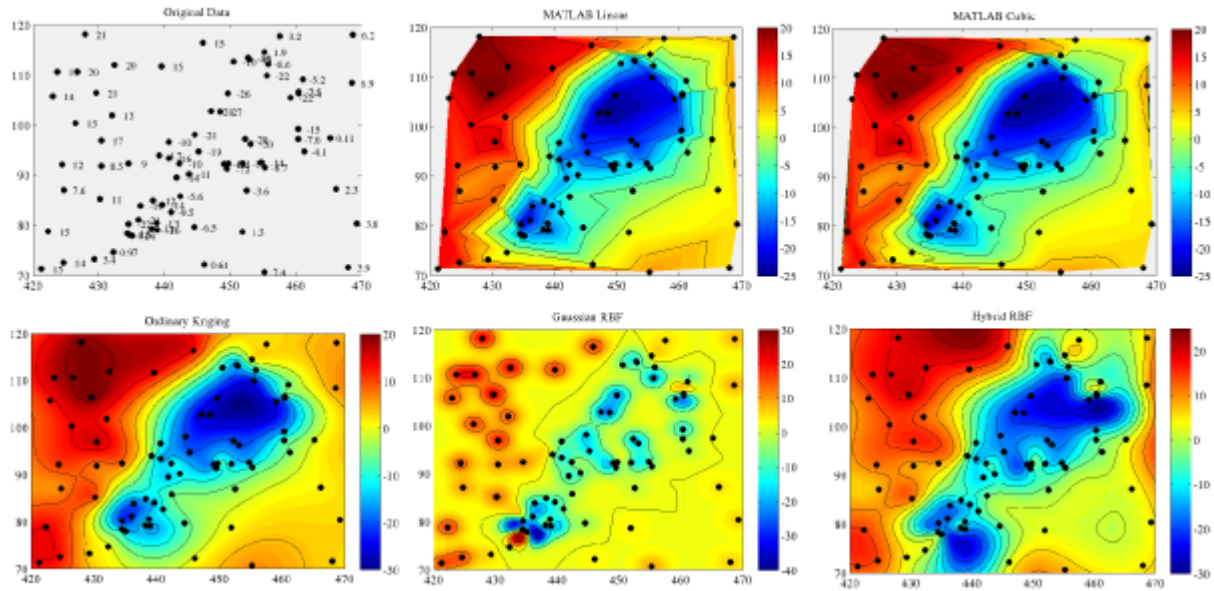
Gitter-freie Interpolation

- Inverse distance weighting (IDW)
- Polynom Interpolation
- Splines und Radial Basis Functions



Gitter-freie Interpolation

- Inverse distance weighting (IDW)
- ~~Polynom Interpolation~~ Polynom Approximation
- Splines und Radial Basis Functions



Inverse Distance Weighting (IDW)

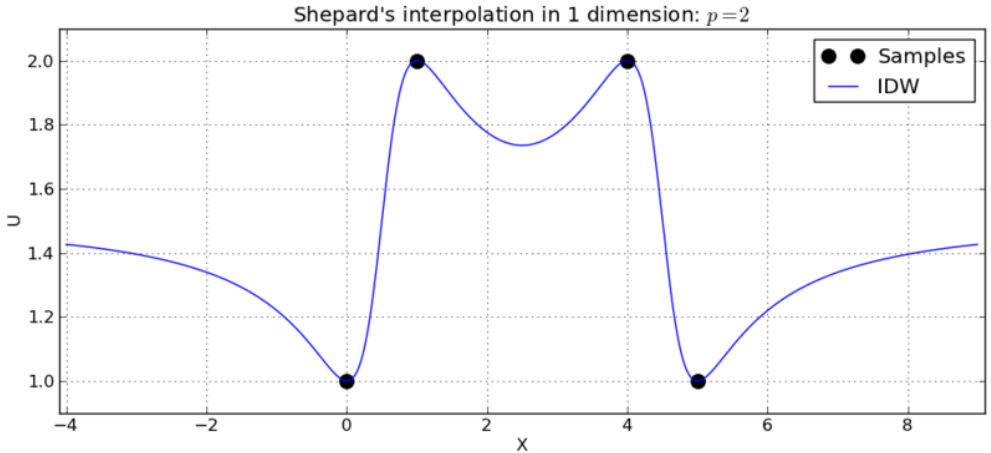
Annahme: Der Wert an einem beliebigen Punkt hängt nur vom Abstand zu den Datenpunkten ab.

Gegeben seien Tupel $(x_i, y_i); i = 1, \dots, N; x_i \in \mathbb{R}^n$. Der unbekannte Wert $f(x)$ an einem beliebigen Punkt $x \in \mathbb{R}^n$ lässt sich vorhersagen mit:

$$f(x) = \frac{\sum_{i=1}^{N} \lambda_i(x) y_i}{\sum_{i=1}^{N} \lambda_i(x)}$$

Mit

$$\lambda_i(x) = \frac{1}{|x - x_i|^p}.$$



Inverse Distance Weighting (IDW)

Annahme: Der Wert an einem beliebigen Punkt hängt nur vom Abstand zu den Datenpunkten ab.

Gegeben seien Tupel $(x_i, y_i); i = 1, \dots, N; x_i \in \mathbb{R}^n$. Der unbekannte Wert $f(x)$ an einem beliebigen Punkt $x \in \mathbb{R}^n$ lässt sich vorhersagen mit:

$$f(x) = \frac{\sum_{i=1}^{N} \lambda_i(x) y_i}{\sum_{i=1}^{N} \lambda_i(x)}$$

Mit

$$\lambda_i(x) = \frac{1}{|x - x_i|^p}.$$

Oder alternativ:

$$f(x) = \sum_{i=1}^{N} \lambda_i(x) y_i \text{ mit } \lambda_i(x) = \frac{\frac{1}{|x-x_i|^p}}{\sum_{j=1}^{N} \frac{1}{|x-x_j|^p}} = \frac{1}{1 + \sum_{j=1, j\neq i}^{N} \frac{|x-x_i|^p}{|x-x_j|^p}}$$

zum Vermeiden des globalen Vorfaktoren …

Inverse Distance Weighting (IDW)

$$f(x) = \sum_{i=1}^{N} \lambda_i(x) y_i \text{ mit } \lambda_i(x) = \frac{\frac{1}{|x-x_i|^p}}{\sum_{j=1}^{N} \frac{1}{|x-x_j|^p}} = \frac{1}{1+\sum_{j=1, j\neq i}^{N} \frac{|x-x_i|^p}{|x-x_j|^p}}$$

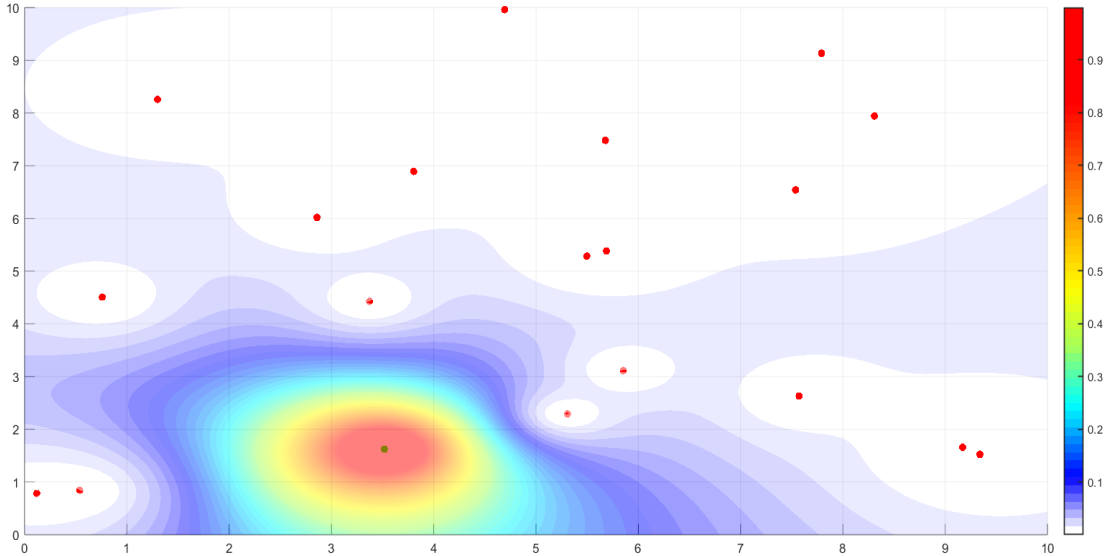


Illustration of weight function λ_1 for various $x \in [0,1]^2$ with parameter $k = 2$.

Inverse Distance Weighting (IDW)

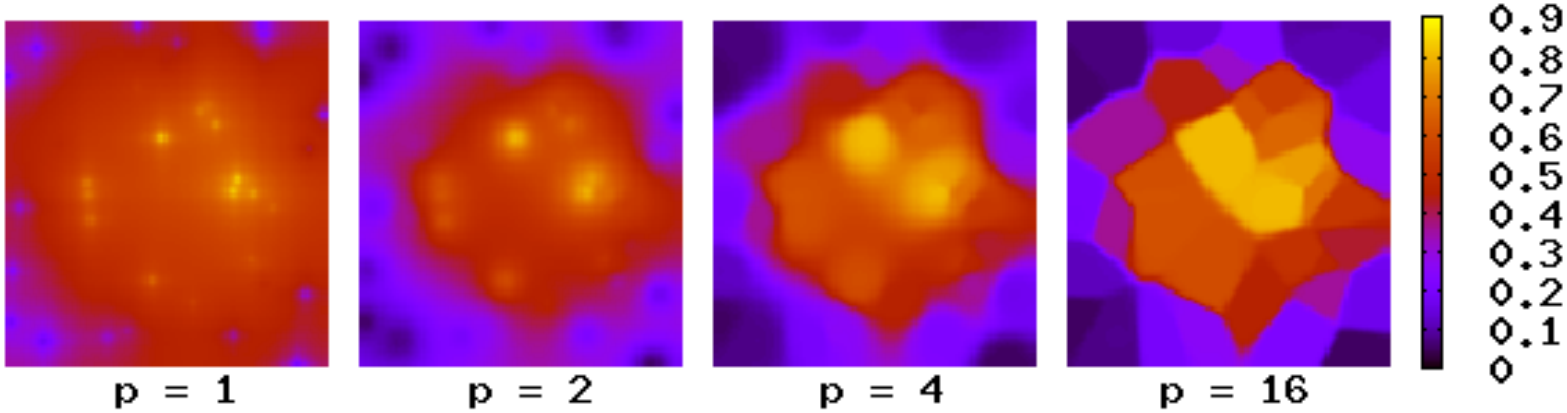
Gegeben seien Tupel $(x_i, y_i); i = 1, \dots, N; x_i \in \mathbb{R}^n$. Der unbekannte Wert $f(x)$ an einem beliebigen Punkt $x \in \mathbb{R}^n$ lässt sich vorhersagen mit:

$$f(x) = \frac{\sum_{i=1}^{N} \lambda_i(x) y_i}{\sum_{i=1}^{N} \lambda_i(x)}$$

Mit

$$\lambda_i(x) = \frac{1}{|x - x_i|^{\boldsymbol{p}}}.$$

Einfluss des Exponenten p als Lokalisierungsparameter:



Inverse Distance Weighting (IDW)

Gegeben seien Tupel $(x_i, y_i); i = 1, \dots, N; x_i \in \mathbb{R}^n$. Der unbekannte Wert $f(x)$ an einem beliebigen Punkt $x \in \mathbb{R}^n$ lässt sich vorhersagen mit:

$$f(x) = \frac{\sum_{i=1}^{N} \lambda_i(x) y_i}{\sum_{i=1}^{N} \lambda_i(x)}$$

Mit

$$\lambda_i(x) = \frac{1}{|x - x_i|^{\boldsymbol{p}}}.$$

Eigenschaften:

- $f(x)$ ist kontinuierlich;
- Globaler Interpolator, welcher alle Daten berücksichtigt;
- Gewichte hängen rein Punktabstand zw. Interpolations- und Datenpunkten ab;

- Extrapolation ist konzeptionell möglich, aber für

$$|x - x_i| \to \infty \; \forall i \Rightarrow f(x) \to \frac{\sum_{i=1}^{N} y_i}{N}$$

Polynom Interpolation

Annahme: Der unbekannte Wert an einem beliebigen Punkt lässt sich über eine Polynomfunktion $f(x) = \sum_{i=0}^{m} \lambda_i x^i$ vorhersagen.

Gegeben seien Tupel $(x_i, y_i); i = 1, \dots, N; x_i \in \mathbb{R}^n$. Sei $f(x) = \sum_{i=0}^{m} \lambda_i x^i$ ein Polynom vom Grad $m = N - 1$, dann kann man die Koeffizienten $\lambda_0, \dots, \lambda_{N-1} \in \mathbb{R}$ so wählen, dass

$$f(x_i) = y_i.$$

Um die Koeffizienten $\lambda_0, \dots, \lambda_m$ zu finden, muss folgendes Gleichungssystem gelöst werden:

$$\mathbf{M}\boldsymbol{\lambda} = \mathbf{y}$$

mit $\boldsymbol{\lambda} = (\lambda_0, \dots, \lambda_m)^{\mathrm{T}}$, $\mathbf{y} = (y_1, \dots, y_N)$ und

$$\mathbf{M} = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_m^2 & \cdots & x_N^m \end{pmatrix}$$

Polynom Interpolation

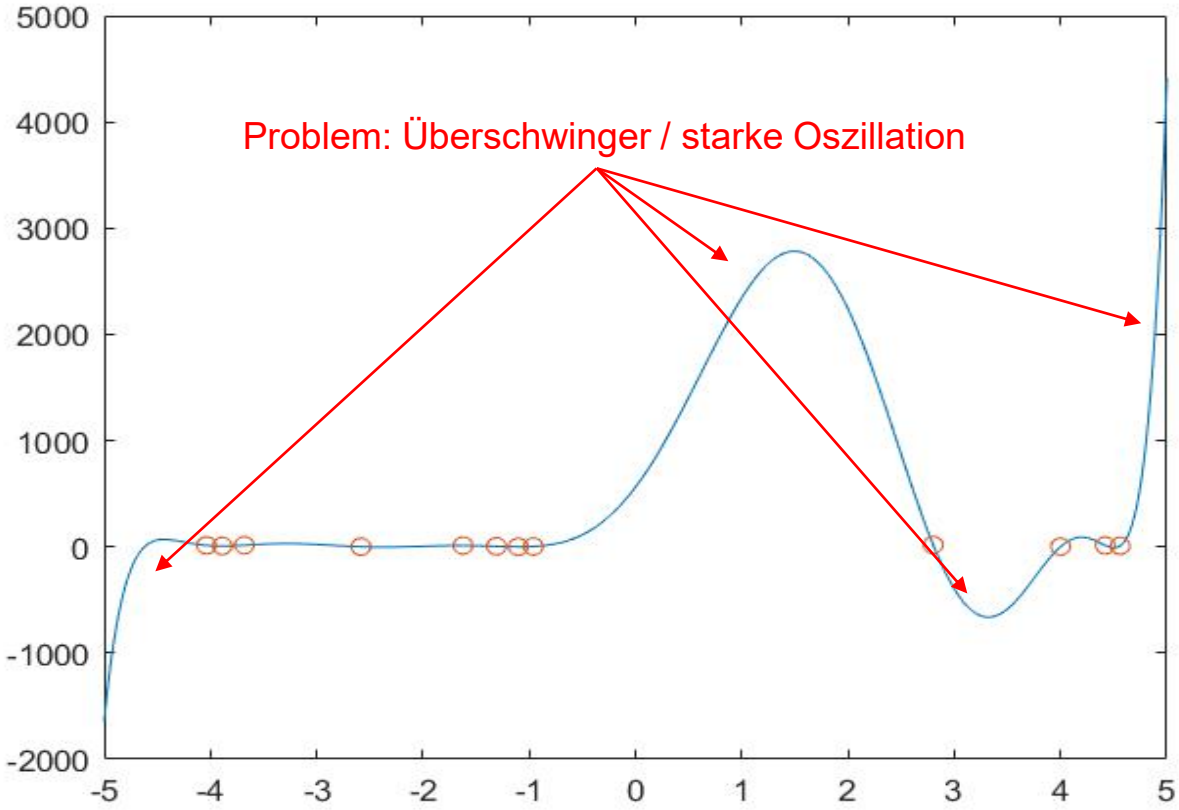
Annah
Polyn

Gege
Grad

Um d

en:

mit λ



Interpolationspolynom für $N = 12$ Datenpunkte und Grad $m = N - 1 = 11$.

Polynom Approximation

Annahme: Der unbekannte Wert an einem beliebigen Punkt lässt sich über eine Polynomfunktion $f(x) = \sum_{i=0}^{m} \lambda_i x^i$ vorhersagen.

Gegeben seien Tupel $(x_i, y_i); i = 1, \dots, N; x_i \in \mathbb{R}^n$ und sei $f(x) = \sum_{i=0}^{m} \lambda_i x^i$ ein Polynom vom Grad m. Die Oszillation des Polynoms außerhalb der Daten kann dadurch reduziert werden, wenn der Grad $m \ll N - 1$ gewählt wird. Die Koeffizienten können nur noch so gewählt werden, dass sie

$$\min_{\lambda_0,\dots,\lambda_m} \sum_{i=1}^{N} |f(x_i) - y_i|^2$$

minimieren. Dieses Verfahren wird als **Approximation** (nicht-exakte Interpolation) bezeichnet, weil im Allgemeinen $f(x_i) = y_i \; \forall i$ **NICHT** erreicht werden kann.
Es gilt nur noch

$$f(x_i) \approx y_i \; \forall i.$$

Eigenschaften:

- $f(x)$ ist ein Polynom vom Grad m (d.h. kontinuierlich und m-mal stetig differenzierbar)
- globaler Interpolator der **alle** (x_i, y_i) mit $1 \leq i \leq N$ berücksichtigt
- für $m \ll N - 1$: Approximations-Polynom $f(x_i) \approx y_i$ mit $1 \leq i \leq N$
- Extrapolation ist konzeptionell möglich

Splines 1-D

Ein **Spline von Grad n** ist eine Funktion, die für eine gegebene Menge Knoten $P = \{x_1, \dots, x_N\} \subset \mathbb{R}$ über jedem offenen Intervall $(x_i, x_{i+1}), (x_n, \infty)$ und $(-\infty, x_1)$ einem Polynom vom Grad n entspricht und deren Ableitungen auf den Knoten $x_1, \dots, x_N$ bis zum Grad $n - 1$ stetig sind.

Der Raum aller Funktionen, die diesen Spline-Eigenschaften entsprechen, wird als $\mathrm{SP}_{P,n}$ bezeichnet.

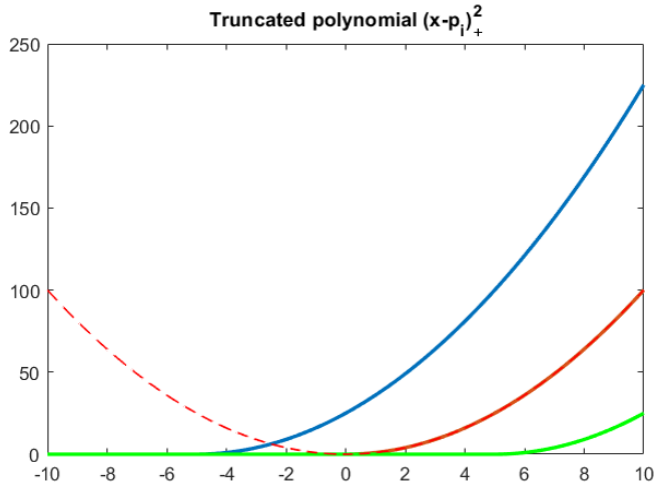
Splines 1-D

Ein **Spline von Grad n** ist eine Funktion, die für eine gegebene Menge Knoten $P = \{x_1, \dots, x_N\} \subset \mathbb{R}$ über jedem offenen Intervall $(x_i, x_{i+1}), (x_n, \infty)$ und $(-\infty, x_1)$ einem Polynom vom Grad n entspricht und deren Ableitungen auf den Knoten $x_1, \dots, x_N$ bis zum Grad $n-1$ stetig sind.

Der Raum aller Funktionen, die diesen Spline-Eigenschaften entsprechen, wird als $\mathrm{SP}_{P,n}$ bezeichnet.

Ein Beispiel für die Element von $\mathrm{SP}_{P,n}$ sind so genannte *verschränkte* (*truncated*) Polynome:

$$(x - p_i)_+^n = \begin{cases} (x - p_i)^n, \text{falls x} \geq p_i, \\ 0, \text{sonst} \end{cases}$$



Splines 1-D

Ein **Spline von Grad n** ist eine Funktion, die für eine gegebene Menge Knoten $P = \{x_1, \dots, x_N\} \subset \mathbb{R}$ über jedem offenen Intervall $(x_i, x_{i+1}), (x_n, \infty)$ und $(-\infty, x_1)$ einem Polynom vom Grad n entspricht und deren Ableitungen auf den Knoten $x_1, \dots, x_N$ bis zum Grad $n-1$ stetig sind.

Der Raum aller Funktionen, die diesen Spline-Eigenschaften entsprechen, wird als $\mathrm{SP}_{P,n}$ bezeichnet.

Ein Beispiel für die Element von $\mathrm{SP}_{P,n}$ sind so genannte *verschränkte* (*truncated*) Polynome:

$$(x - p_i)_+^n = \begin{cases} (x - p_i)^n, \text{falls x} \geq p_i, \\ 0, \text{sonst} \end{cases}$$

Die n-te Ableitung $S^{(n)}$ eines Splines aus $\mathrm{SP}_{P,n}$ führt zu einer Stufenfunktion mit Sprüngen an x_i.

Splines 1-D

Die Dimension von $\mathrm{SP}_{P,n}$ ist

$$\dim\left(\mathrm{SP}_{P,n}\right) = n + N + 1,$$

dies kann wie folgt interpretiert werden: Auf jedem Intervall (x_i, x_{i+1}) liegen $n + 1$ Freiheitsgrade. Durch die Ableitungsbedingung an den Knoten geht dort ein n Freiheitsgrade verloren:

$$\dim\left(\mathrm{SP}_{P,n}\right) = (N + 1)(n + 1) - nN = n + N + 1$$

Splines 1-D

Die Dimension von $\mathrm{SP}_{P,n}$ ist

$$\dim\left(\mathrm{SP}_{P,n}\right) = n + N + 1,$$

dies kann wie folgt interpretiert werden: Auf jedem Intervall (x_i, x_{i+1}) liegen $n + 1$ Freiheitsgrade. Durch die Ableitungsbedingung an den Knoten geht dort ein n Freiheitsgrade verloren:

$$\dim\left(\mathrm{SP}_{P,n}\right) = (N + 1)(n + 1) - nN = n + N + 1$$

Ein Spline S kann daher wie folgt ausgedrückt werden:

$$S(x) = \sum_{i=1}^{N} \lambda_i (x - x_i)_+^n + \sum_{j=0}^{n} \lambda_{j+N+1} x^j$$

Die Gewichte $\lambda_1, \dots, \lambda_{n+N+1}$ werden über ein lineares Gleichungssystem bestimmt, ähnlich der Polynom-Interpolation / -approximation,

Splines 1-D

Die Dimension von $\mathrm{SP}_{P,n}$ ist

$$\dim\left(\mathrm{SP}_{P,n}\right) = n + N + 1,$$

dies kann wie folgt interpretiert werden: Auf jedem Intervall (x_i, x_{i+1}) liegen $n + 1$ Freiheitsgrade. Durch die Ableitungsbedingung an den Knoten geht dort ein n Freiheitsgrade verloren:

$$\dim\left(\mathrm{SP}_{P,n}\right) = (N + 1)(n + 1) - nN = n + N + 1$$

Ein Spline S kann daher wie folgt ausgedrückt werden:

$$S(x) = \sum_{i=1}^{N} \lambda_i (x - x_i)_+^n + \sum_{j=0}^{n} \lambda_{j+N+1} x^j$$

Die Gewichte $\lambda_1, \dots, \lambda_{n+N+1}$ werden über ein lineares Gleichungssystem bestimmt, ähnlich der Polynom-Interpolation / -approximation.

Zumeist liegen nur ein Messwert y_i pro Knoten x_i und damit nur N Randbedingungen vor. Um das o.g. Problem lösen zu können, werden $n + 1$ zusätzliche Randbedingungen benötigt. Im Fall von $n = 2m - 1$ könnten diese z.B. sein:

$$S^{(k)}(x_1) = S^{(k)}(x_N) = 0, k = m, \dots, n - 1$$

Splines, die diese Bedingung erfüllen werden als **natürliche Splines** bezeichnet, der zugehörige Raum $\mathrm{SPN}_{P,n}$.

Splines 1-D

Warum sind Splines so nützlich?

Liegen die Werte y_i an den Punkten x_i mit $i = 1, \dots, N$ vor, so gilt für den Spline $S \in \mathrm{SPN}_{P,n}$, der

$$S(x_i) = y_i, i = 1, \dots, N$$

interpoliert, dass er folgenden Ausdruck minimiert:

$$\min_{f \text{ interpolates } P} \int_{x_i}^{x_N} |f^n(x)|^2 dx$$

(Minimierung der n-ten Ableitung der Interpolationsfunktion).

Diese Splines werden auch als **kubische Splines** bezeichnet und kombinieren den einfachen Charakter von Polynominterpolation mit zusätzlicher Stabilität und geringerer Oszillation.

Splines 1-D

Warum sind Splines so nützlich?

Liegen die Werte y_i an den Punkten x_i mit $i = 1, \dots, N$ vor, so gilt für den Spline $S \in \mathrm{SPN}_{P,n}$, der

$$S(x_i) = y_i, i = 1, \dots, N$$

interpoliert, dass er folgenden Ausdruck minimiert:

$$\min_{f \text{ interpolates } P} \int_{x_i}^{x_N} |f^n(x)|^2 dx$$

(Minimierung der n-ten Ableitung der Interpolationsfunktion).

Diese Splines werden auch als **kubische Splines** bezeichnet und kombinieren den einfachen Charakter von Polynominterpolation mit zusätzlicher Stabilität und geringerer Oszillation.

Eigenschaften:

- $f(x)$ ist kontinuierlich und min. $n - 1$-mal stetig differenzierbar
- Globaler Interpolator
- Extrapolation ist konzeptuell möglich (hier können dann aber Oszillationen auftreten)

Spline n-D / Radial Basis Functions (RBFs)

Wenn man annimmt, dass ein regelmäßiges Gitter von Knoten vorliegt, lässt sich der 1D-Ansatz zu Splines auf beliebige Dimensionen erweitern. Im Allgemeinen liegen Daten aber nicht auf einem solchen Gitter vor. Hier müssen andere Ansätze verwendet werden. Eine Option hier sind so genannte radiale Basisfunktionen (*radial basis functions* / RBF).

Liegen die Werte y_i an den Punkten $x_i \in \mathbb{R}^2$ mit $i = 1, \dots, N$ vor, dann kann man mit einem so genannten **thin plate spline** der Form

$$S(x) = \sum_{i=1}^{N} \lambda_i \Phi(|x - x_i|)$$

mit $\Phi(\mathrm{r}) = \mathrm{r}^2 \ln(\mathrm{r})$, welche

$$S(x_i) = y_i, i = 1, \dots, N$$

interpoliert, ist die Interpolationsfunktion, welche folgenden Ausdruck minimiert:

$$\min_{f \text{ interpoliert } P} \int_D \left|\partial_{x_1}^2 f(x)^2\right| + \left|\partial_{x_1}\partial_{x_2} f(x)^2\right| + \left|\partial_{x_2}^2 f(x)^2\right| dx\,.$$

Spline n-D / Radial Basis Functions (RBFs)

Jede Funktion $\Phi: [0, \infty) \rightarrow \mathbb{R}$ die zu einer Funktion der Form $\phi(x) = \Phi(|\mathbf{x}|)$ führt, die nur von der Länge eines Vektors abhängt, kann als **radiale Basisfunktion** bezeichnet werden. Typischerweise sind weitere „gutartige“ Eigenschaften wie z.B. positive Definiertheit, gewünscht. Eine Approximationsfunktion basierend auf diesem Ansatz ist immer der Form

$$S(x) = \sum_{i=1}^{N} \lambda_i \phi_i(x)$$

mit $\phi_i(x) = \Phi(|x - x_i|)$ und $S(x_i) = y_i, i = 1, \dots, N$.

Spline n-D / Radial Basis Functions (RBFs)

Jede Funktion $\Phi: [0, \infty) \rightarrow \mathbb{R}$ die zu einer Funktion der Form $\phi(x) = \Phi(|\mathbf{x}|)$ führt, die nur von der Länge eines Vektors abhängt, kann als **radiale Basisfunktion** bezeichnet werden. Typischerweise sind weitere „gutartige“ Eigenschaften wie z.B. positive Definiertheit, gewünscht. Eine Approximationsfunktion basierend auf diesem Ansatz ist immer der Form

$$S(x) = \sum_{i=1}^{N} \lambda_i \phi_i(x)$$

mit $\phi_i(x) = \Phi(|x - x_i|)$ und $S(x_i) = y_i, i = 1, \dots, N$.

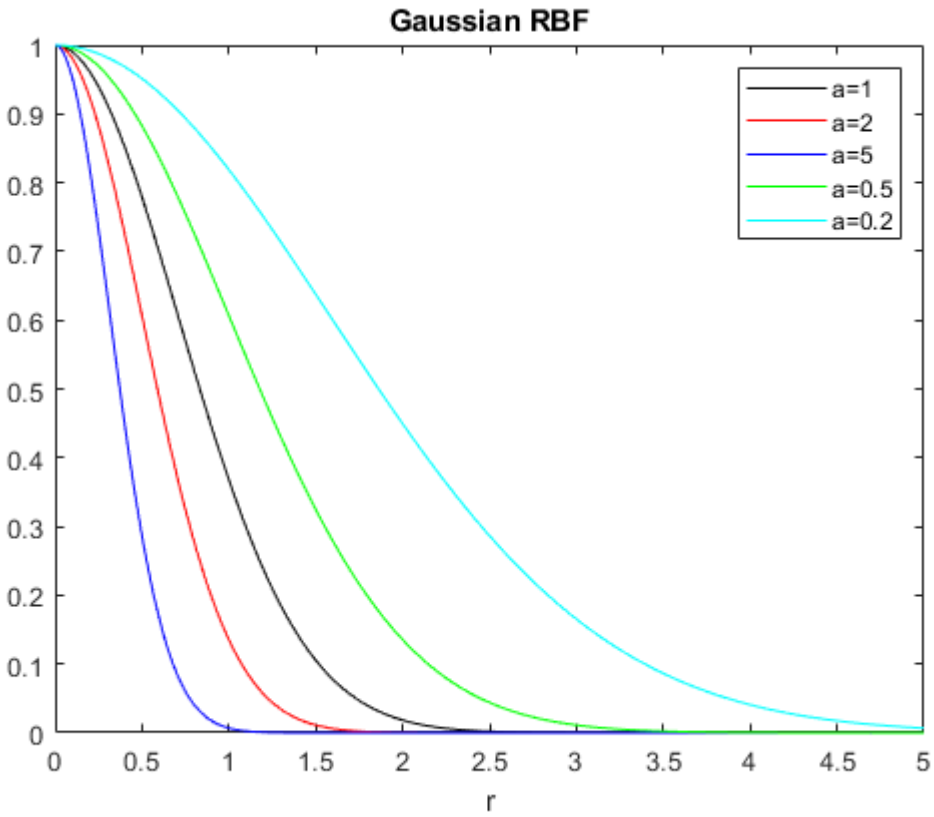
Häufige verwendete positiv definite Basisfunktionen sind z.B.

$$\Phi(r) = e^{-a^2 r^2}, (\text{Gaussian})$$
$$\Phi(r) = (r^2 + a^2)^{\frac{1}{2}}, (\text{Multiquadric})$$
$$\Phi(r) = \frac{1}{r^2 + a^2}, (\text{Inverse Quadric})$$
$$\Phi(\mathrm{r}) = \mathrm{r}^2 \ln(\mathrm{r}), (\text{thin plate})$$

Andere mögliche RBFs (z.B. mit kompaktem Support) sind komplexer.

Spline n-D / Radial Basis Functions (RBFs)

Jede Funktion $\Phi: [0, \infty) \rightarrow \mathbb{R}$ die zu einer Funktion der Form $\phi(x) = \Phi(|\mathbf{x}|)$ führt, die nur von der Länge
werden.
Typischer
ertheit, gewünscht.
Eine Appr



mit $\phi_i(x)$

Häufige v

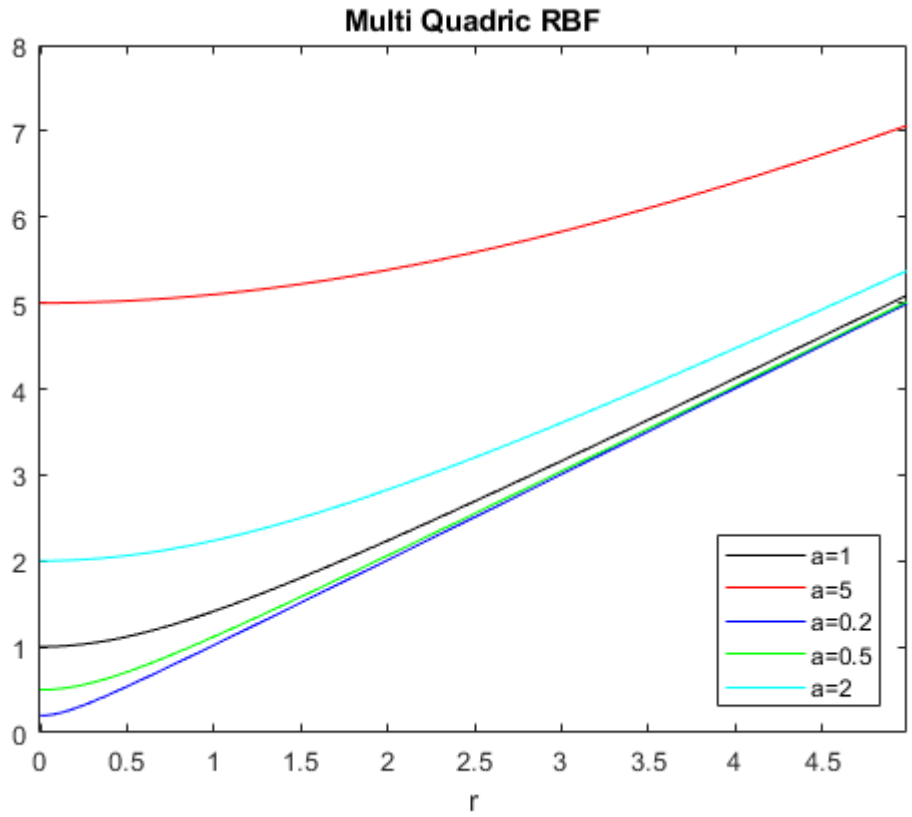
Andere m

Spline n-D / Radial Basis Functions (RBFs)

Jede Funktion $\Phi: [0, \infty) \rightarrow \mathbb{R}$ die zu einer Funktion der Form $\phi(x) = \Phi(|\mathbf{x}|)$ führt, die nur von der Länge eir rden.

Typischerweis eit, gewünscht.

Eine Approxir



mit $\phi_i(x) = \Phi$

Häufige verw

Andere mögli

Spline n-D / Radial Basis Functions (RBFs)

Zur Bestimmung der notwendigen Koeffizienten $\lambda_1, \dots, \lambda_m$ muss wieder ein lineares Gleichungssystem der Form

$$\mathbf{M}\boldsymbol{\lambda} = \mathbf{y}$$

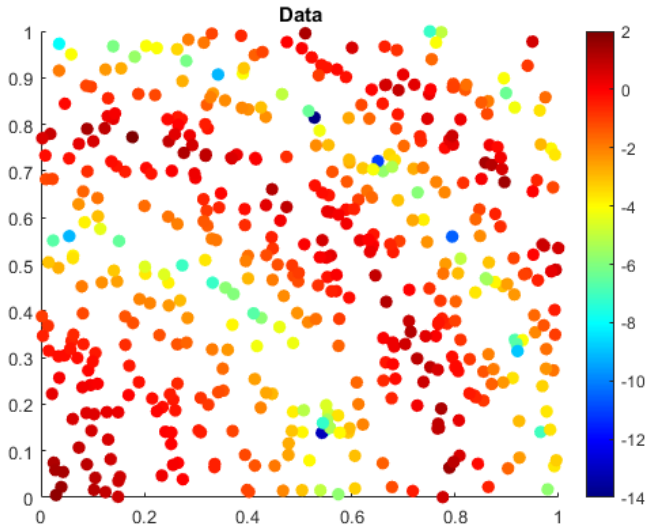
mit $\boldsymbol{\lambda} = (\lambda_0, \dots, \lambda_m)^{\mathrm{T}}$, $\mathbf{y} = (y_1, \dots, y_N)$ und

$$\mathbf{M} = \begin{pmatrix} \Phi(0) & \Phi(|x_1 - x_2|) & \Phi(|x_1 - x_3|) & \cdots & \Phi(|x_1 - x_N|) \\ \Phi(|x_2 - x_1|) & \Phi(0) & \Phi(|x_2 - x_3|) & \cdots & x_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi(|x_N - x_1|) & \Phi(|x_N - x_2|) & \Phi(|x_N - x_3|) & \cdots & \Phi(0) \end{pmatrix}$$

Die (eindeutige) Lösbarkeit und damit die Qualität der Interpolaton hängen sehr stark vom gewählten Typ der RBF ab.

Spline n-D / Radial Basis Functions (RBFs)

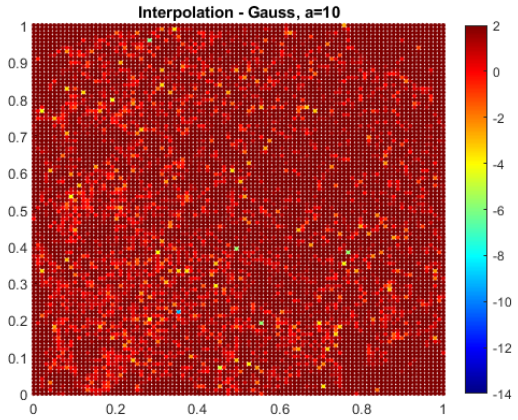
Zur Bestimmung der r … ler ein lineares Gleichungssystem de …

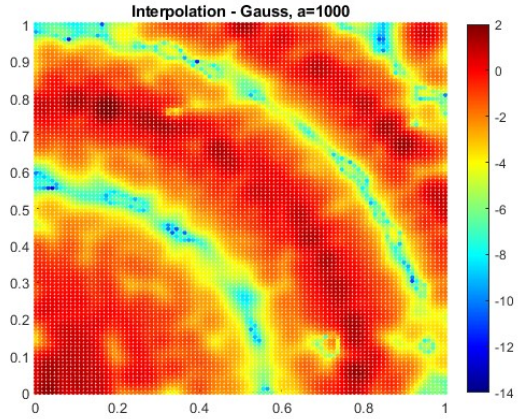


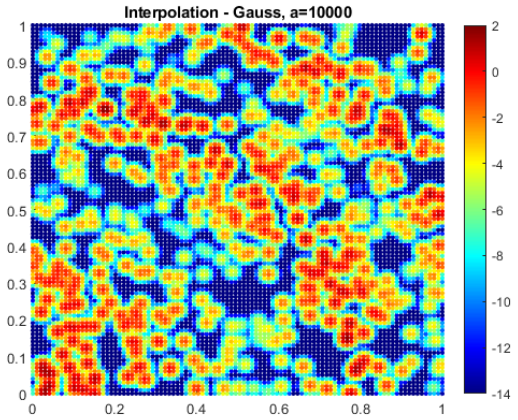
mit $\boldsymbol{\lambda} = (\lambda_0, \ldots, \lambda_m)^{\mathrm{T}}$, $\mathbf{y}$ …

$$\mathbf{M} = \begin{pmatrix} \Phi((& & & \Phi(|x_1 - x_N|) \\ \Phi(|x_2 - & & & x_2^m \\ \vdots & & & \vdots \\ \Phi(|x_N - & & & \Phi(0) \end{pmatrix}$$

Die (eindeutige) Lösbarkeit und damit die Qualität der Interpolaton hängen sehr stark







Spline n-D / Radial Basis Functions (RBFs)

Zur Bestimmung der notwendigen Koeffizienten $\lambda_1, \dots, \lambda_m$ muss wieder ein lineares Gleichungssystem der Form

$$\mathbf{M}\boldsymbol{\lambda} = \mathbf{y}$$

mit $\boldsymbol{\lambda} = (\lambda_0, \dots, \lambda_m)^{\mathrm{T}}$, $\mathbf{y} = (y_1, \dots, y_N)$ und

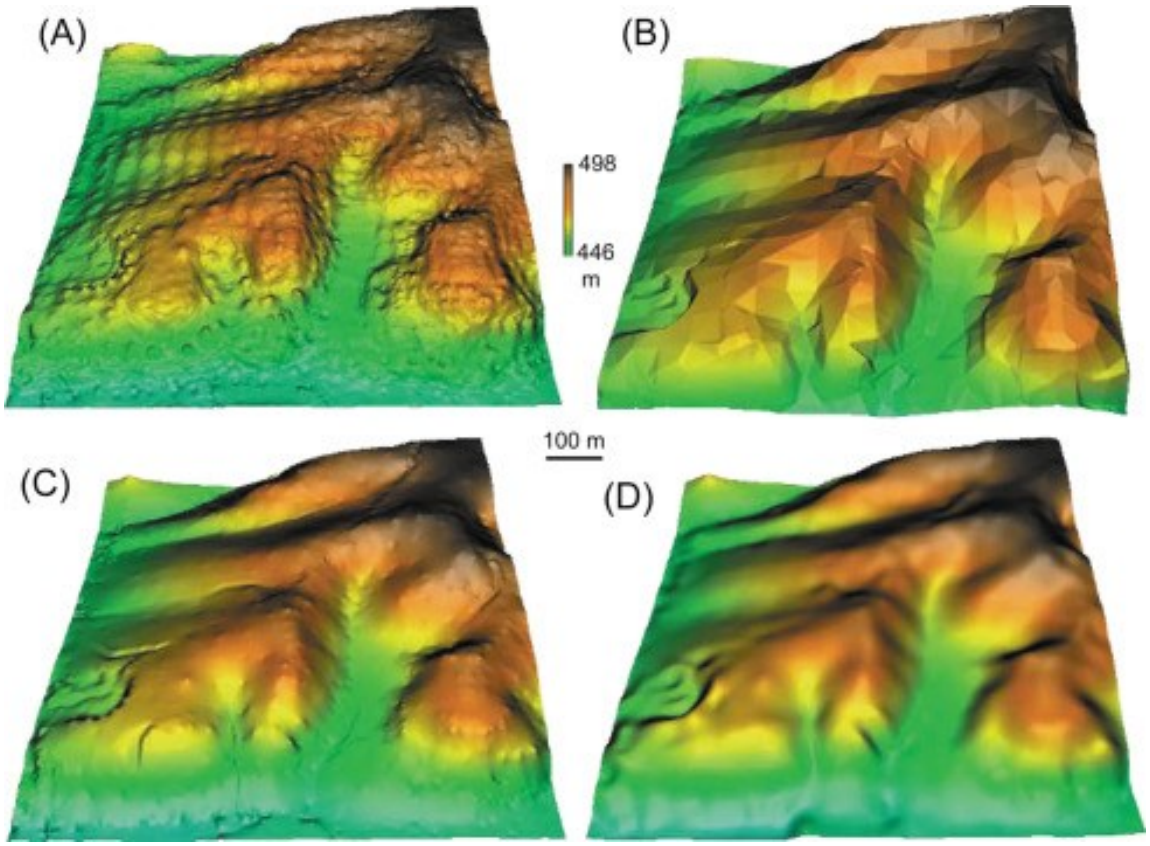
$$\mathbf{M} = \begin{pmatrix} \Phi(0) & \Phi(|x_1 - x_2|) & \Phi(|x_1 - x_3|) & \cdots & \Phi(|x_1 - x_N|) \\ \Phi(|x_2 - x_1|) & \Phi(0) & \Phi(|x_2 - x_3|) & \cdots & x_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi(|x_N - x_1|) & \Phi(|x_N - x_2|) & \Phi(|x_N - x_3|) & \cdots & \Phi(0) \end{pmatrix}$$

Die (eindeutige) Lösbarkeit und damit die Qualität der Interpolaton hängen sehr stark vom gewählten Typ der RBF ab.

Eigenschaften:

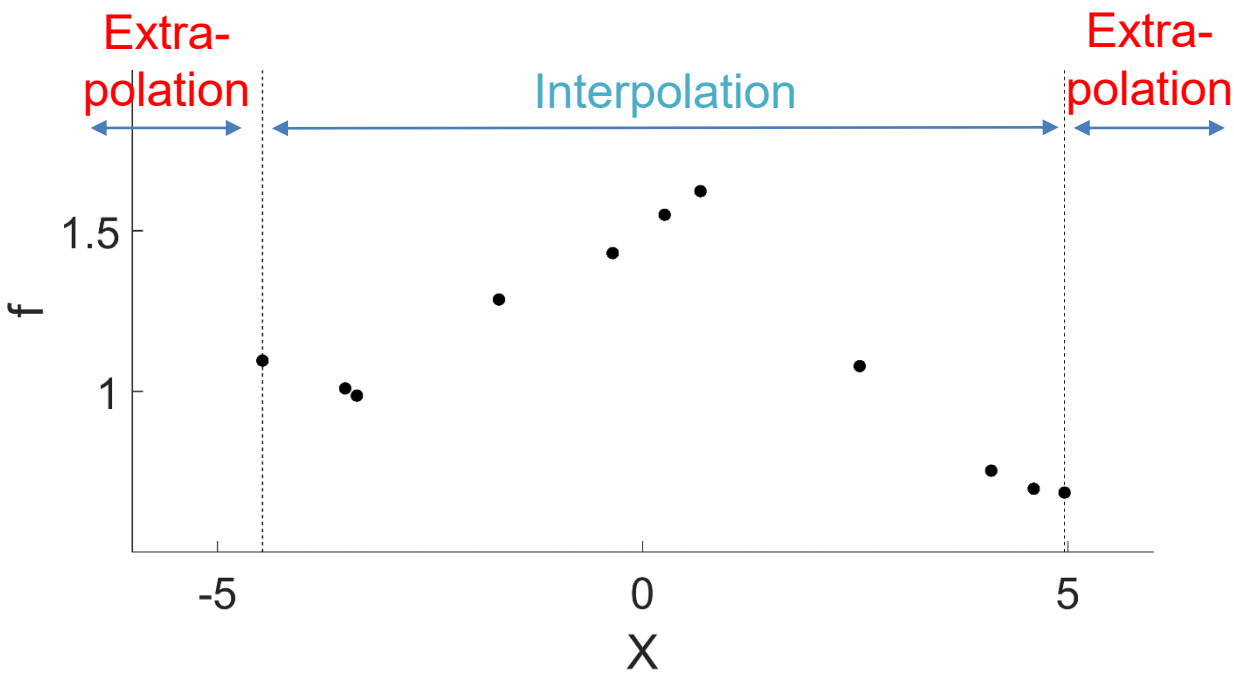
- $f(x)$ ist stetig und differenzierbar
- Globaler Interpolatior
- Extrapolation ist konzeptionell möglich

Example IDW, Lin. Interp. on Triangulation, Splines



Interpolation vs Extrapolation

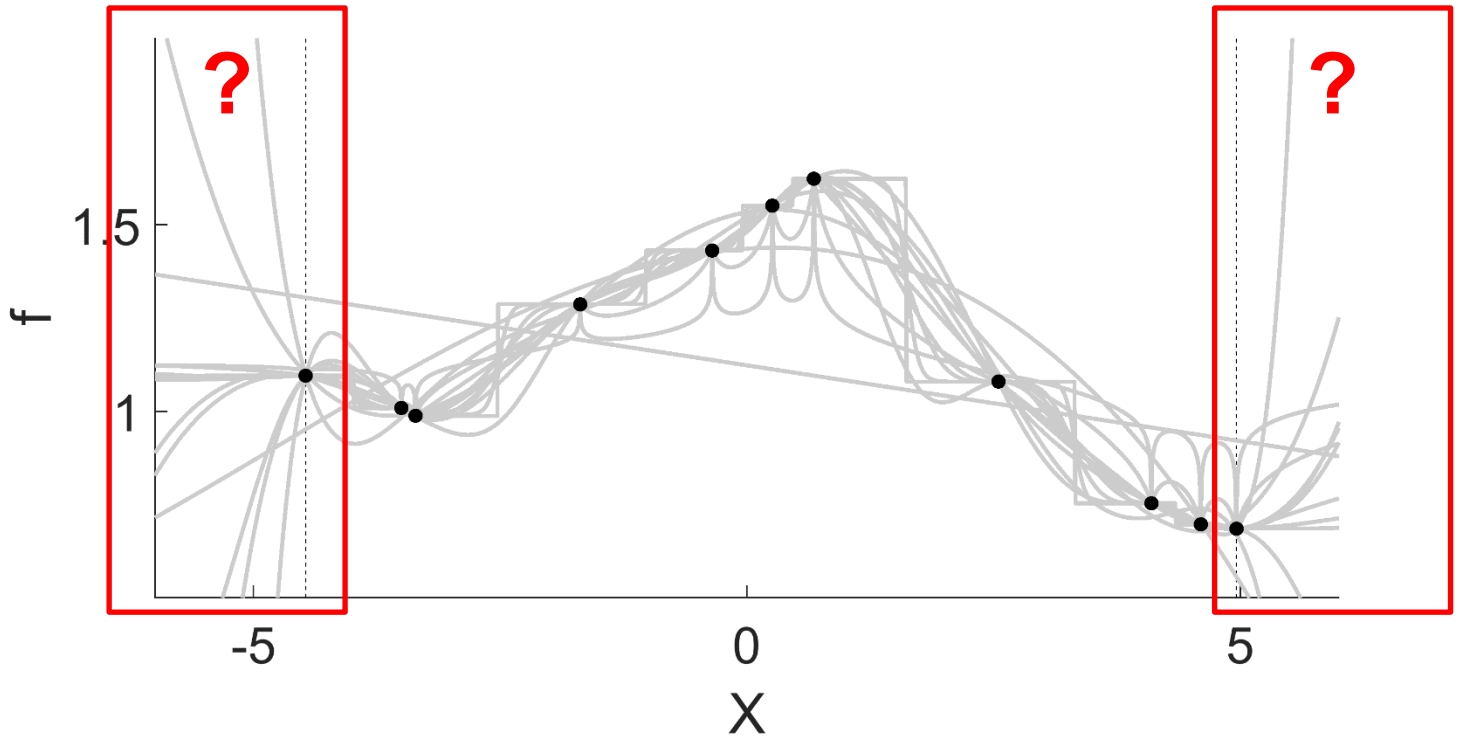
- **Inter**polation: Vorhersage **innerhalb** der Daten / **zwischen** den Datenpunkten
- **Extra**polation: Vorhersage **außerhalb** der Daten (zumeist außerhalb der konvexen Hülle)



Interpolation vs Extrapolation

Was ist das Problem bei Extrapolation?

- Vorhersage basierend auf ungenügender Datengrundlage
- **Vorhersage rein Modell-basiert, zumeist nicht plausibel**



Interpolation

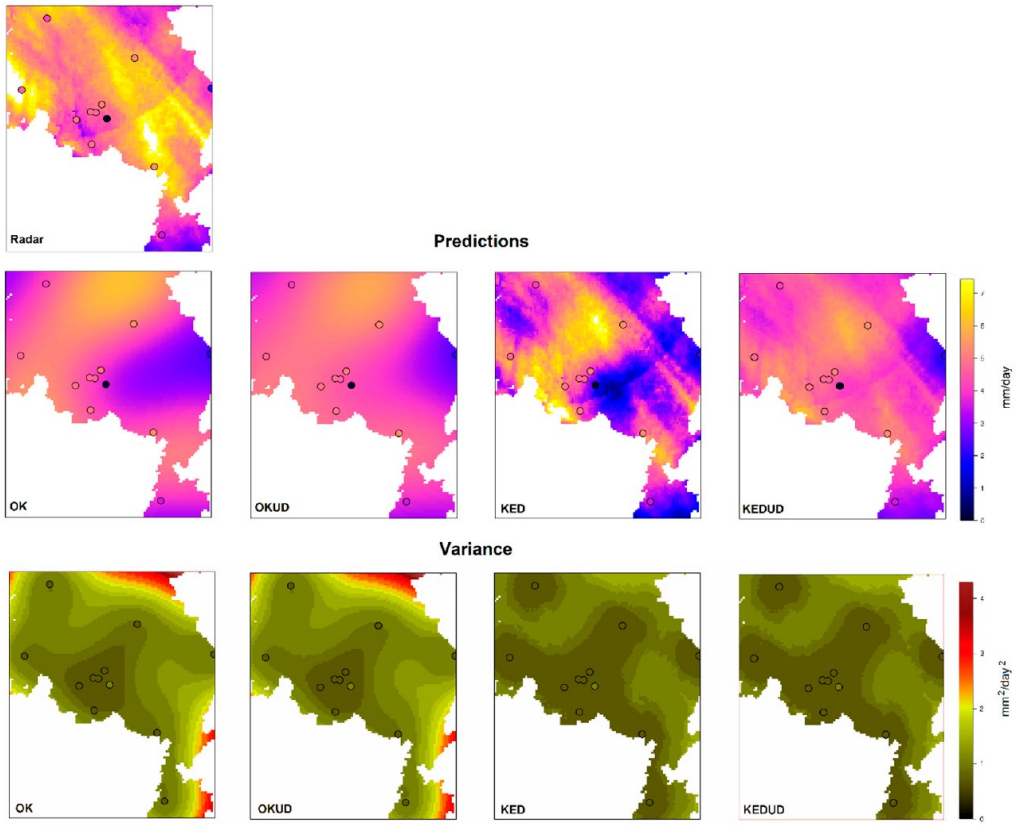
Wie kann man entscheiden, welches Verfahren man nutzen sollte …

Folgende Fragen sollten berücksichtigt werden:

- Ist lokale oder globale Interpolation notwendig?
- Ist Extrapolation notwendig?
- Welche mathematischen Eigenschaften hat der Zielparameter?
- Gibt es Zusatzinformationen, welche durch die Interpolation berücksichtigt werden müssen?
- …

Statistical Methods

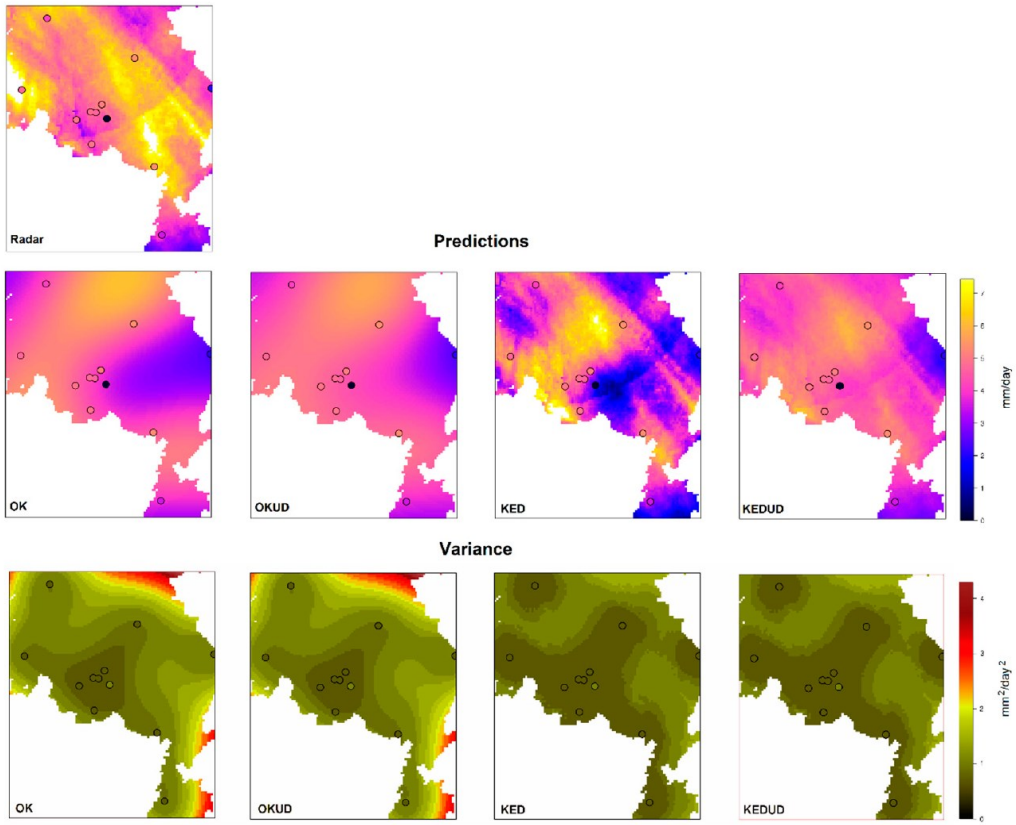
- Kriging (Simple, Ordinary, Universal)
- Conditional Simulation



Cecinati et al.: Considering Rain Gauge Uncertainty Using Kriging for Uncertain Data, Atmosphere (2018)

Statistical Methods

- Kriging (Simple, ~~Ordinary, Universal~~)
- ~~Conditional Simulation~~



Cecinati et al.: Considering Rain Gauge Uncertainty Using Kriging for Uncertain Data, Atmosphere (2018)

Kriging

Der Messwert Z_i an einer Position $x_i \in \mathbb{R}^n$ kann als Zufallsvariable mit **Erwartungswert**

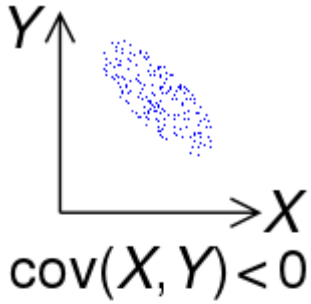
$$E[Z_i] = \int_{\Omega} z d\mu_{Z_i}(z)$$

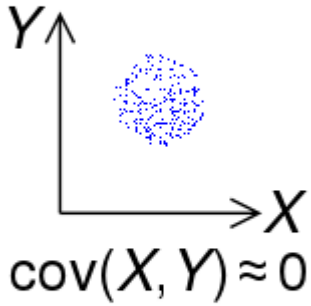
und **Varianz**

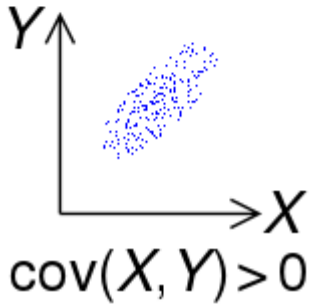
$$\mathrm{Var}(Z_i) = E[(Z_i - E[Z_i])^2].$$

Die **Kovarianz** zwischen zwei Zufallsvariablen ist gegeben durch

$$\mathrm{Cov}(Z_i, Z_j) = E[(Z_i - E[Z_i])(Z_j - E[Z_j])].$$







Kriging

Der Messwert Z_i an einer Position $x_i \in \mathbb{R}^n$ kann als Zufallsvariable mit **Erwartungswert**

$$E[Z_i] = \int_{\Omega} z d\mu_{Z_i}(z)$$

und **Varianz**

$$\mathrm{Var}(Z_i) = E[(Z_i - E[Z_i])^2].$$

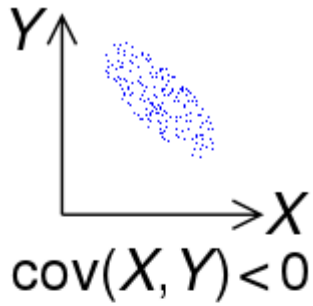
Die **Kovarianz** zwischen zwei Zufallsvariablen ist gegeben durch

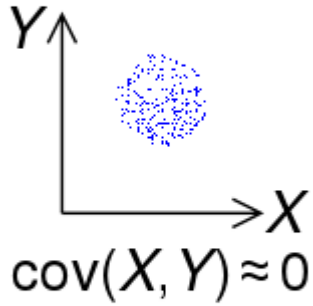
$$\mathrm{Cov}(Z_i, Z_j) = E[(Z_i - E[Z_i])(Z_j - E[Z_j])].$$

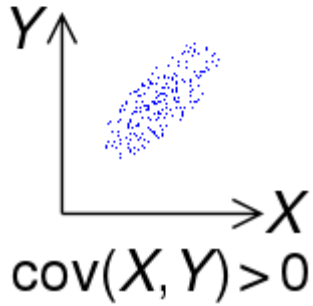
In der Geostatistik ist die so genannte **Stationaritäts*-Annahme** von entscheidender Bedeutung:

$$\boldsymbol{E[Z(x)] = m}, \qquad \mathrm{Cov}(\boldsymbol{Z(x+h), Z(x)}) = \mathrm{Cov}(\boldsymbol{h}).$$

Die Kovarianz hängt nur vom Inkrement h zwischen Positionen und nicht von den Positionen selbst ab.







** stationarity, homogenity*

Kriging

Die Näherung Z^* von Z ist gegeben durch

$$Z^*(x) = \lambda_0 + \sum_{i=1}^{N} \lambda_i(x) Z_i \,,$$

mit den unbekannten Gewichten $\lambda_i(x)$ so dass

$$Z^*(x_i) = z_i, \qquad i = 1, \ldots, N.$$

Kriging

Die Näherung Z^* von Z ist gegeben durch

$$Z^*(x) = \lambda_0 + \sum_{i=1}^{N} \lambda_i(x) Z_i \,,$$

mit den unbekannten Gewichten $\lambda_i(x)$ so dass

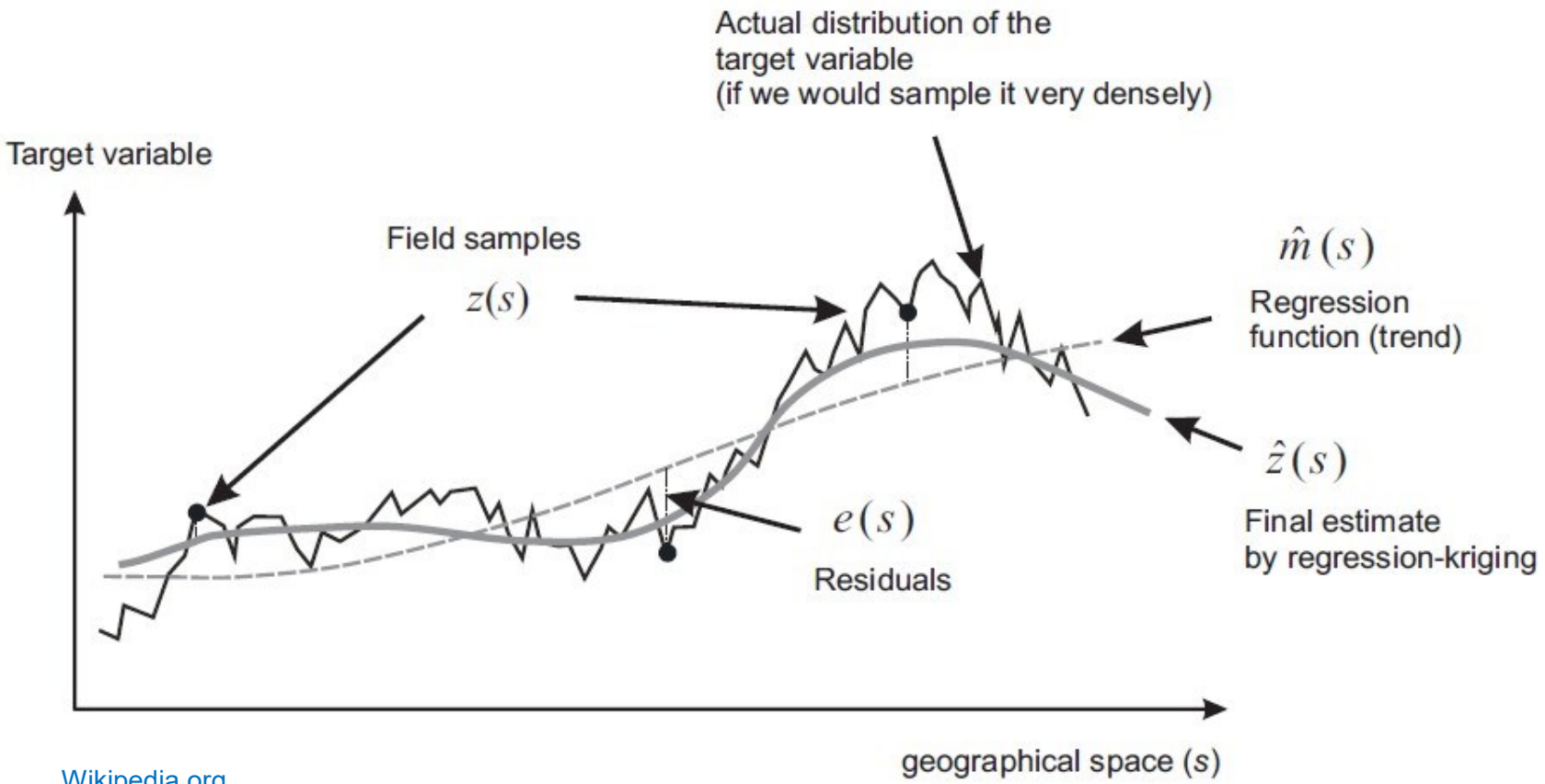
$$Z^*(x_i) = z_i, \qquad i = 1, \dots, N.$$

Welche Annahmen kann man zur Bestimmung der Gewichte treffen?

Unbiasedness (unverfälscht, ideal): $E[Z^*(x)] = E[Z(x)]$

Varianz-Minimierung: $\min_{Z*} \mathrm{Var}(Z^*(x) - Z(x))$

Kriging



Wikipedia.org

Kriging

Simple Kriging: Erwartungswert $m_0 = E(Z(x))$ ist bekannt; Unbiasedness-Annahme führt zu

$$\lambda_0 = m_0 - \sum_{i=1}^{N} \lambda_i m_0 \,.$$

Kriging

Simple Kriging: Erwartungswert $m_0 = E(Z(x))$ ist bekannt; Unbiasedness-Annahme führt zu

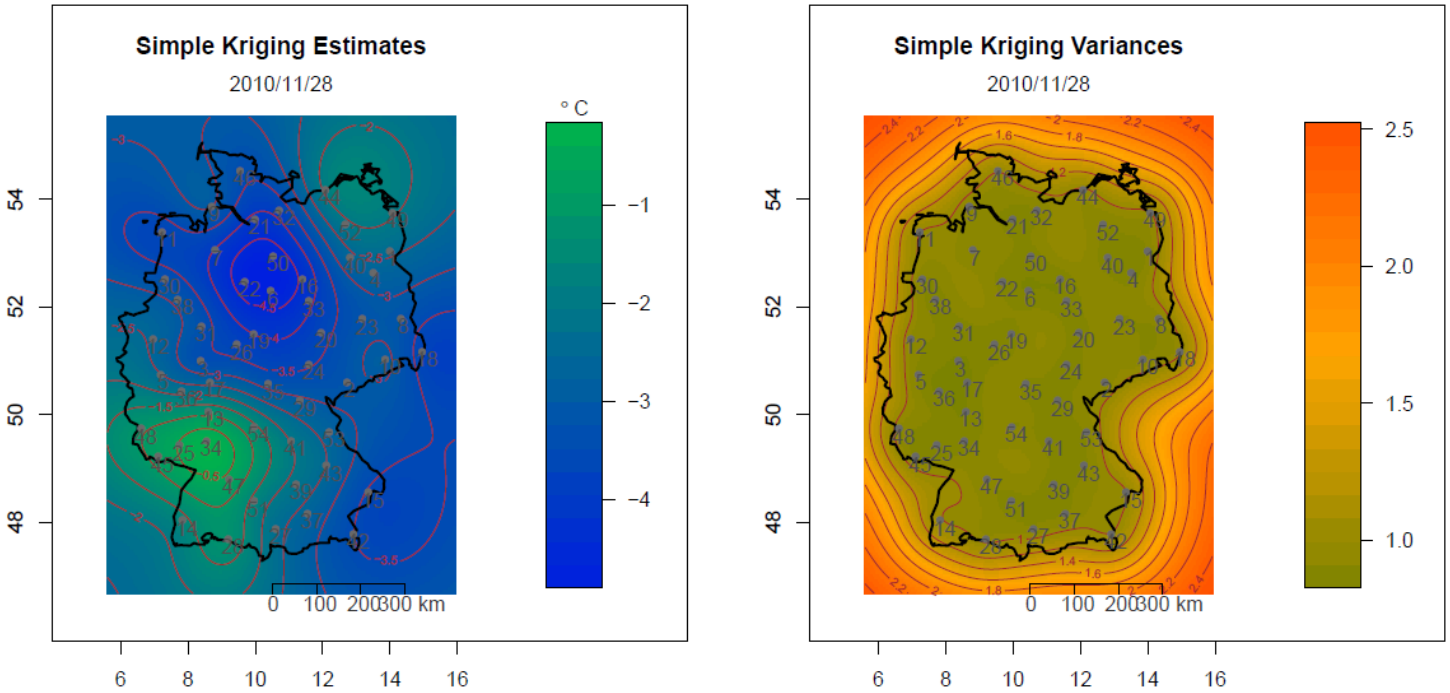
$$\lambda_0 = m_0 - \sum_{i=1}^{N} \lambda_i m_0 \,.$$

Varianz-Minimierung führt zu:

$$\sum_{i=1}^{N} \lambda_i(x) \mathrm{Cov}(Z_i, Z_k) = \mathrm{Cov}\big(Z_i, Z(x)\big), \qquad k = 1, \dots, N$$

Aufgrund der Stationaritäts-Annahme gilt: $\mathrm{Cov}\big(Z_i, Z(x)\big) = \mathrm{Cov}(x_i - x)$ → Ähnlichkeit zu radialen Basisfunktionen.

Kriging



Example from the Bachelor thesis ”Kriging methods in spatial statistics”, Andreas Lichtenstern, TU München

Kriging

Im Allgemeinen ist $\mathrm{Cov}(Z(x_i), Z(x_j))$ nicht bekannt und muss geschätzt oder modelliert werden.

Dafür spielt das so genannte **Variogramm** eine entscheidende Rolle:

$$\gamma(h) = \frac{1}{2}\mathrm{Var}\big(Z(x) - Z(x+h)\big) = C(0) - C(h).$$

Das Variogramm kann unter bestimmten Bedingungen aus den Daten geschätzt werden.

Kriging

Im Allgemeinen ist $\mathrm{Cov}(Z(x_i), Z(x_j))$ nicht bekannt und muss geschätzt oder modelliert werden.

Dafür spielt das so genannte **Variogramm** eine entscheidende Rolle:

$$\gamma(h) = \frac{1}{2}\mathrm{Var}\big(Z(x) - Z(x+h)\big) = C(0) - C(h).$$

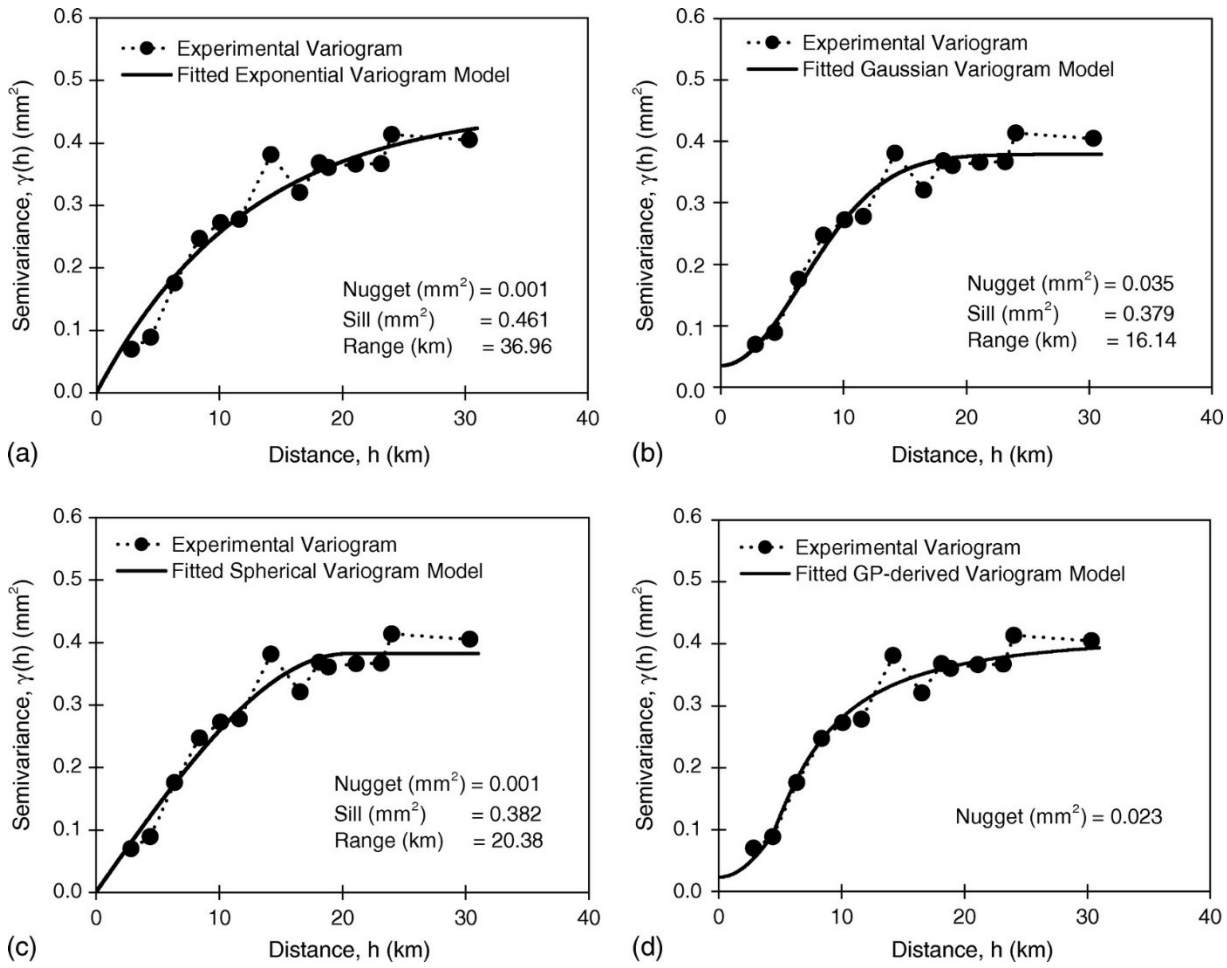
Das Variogramm kann unter bestimmten Bedingungen aus den Daten geschätzt werden.

Empirisches Variogramm:

$$\hat{\gamma}(h) = \frac{1}{N_h} \sum_{x_i - x_j = h} \big(Z_i - Z_j\big)^2.$$

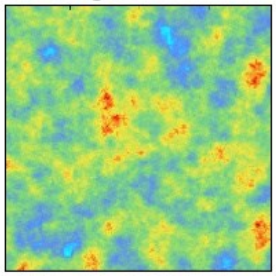
An das empirischen Variogramms kann ein analytisches Variogrammmodell angepasst werden.
Diese Modellfunktion muss dabei bestimmte Eigenschaften erfüllen.
Sie muss z.B. semi-positiv definit sein, d. H. alle Eigenwerte sind größer gleich Null.

Kriging

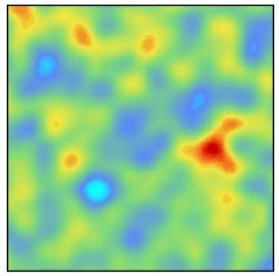

Experimental Variogram
Fitted Exponential Variogram Model
Nugget (mm²) = 0.001
Sill (mm²) = 0.461
Range (km) = 36.96
Semivariance, γ(h) (mm²)
Distance, h (km)
(a)
Experimental Variogram
Fitted Gaussian Variogram Model
Nugget (mm²) = 0.035
Sill (mm²) = 0.379
Range (km) = 16.14
Semivariance, γ(h) (mm²)
Distance, h (km)
(b)
Experimental Variogram
Fitted Spherical Variogram Model
Nugget (mm²) = 0.001
Sill (mm²) = 0.382
Range (km) = 20.38
Semivariance, γ(h) (mm²)
Distance, h (km)
(c)
Experimental Variogram
Fitted GP-derived Variogram Model
Nugget (mm²) = 0.023
Semivariance, γ(h) (mm²)
Distance, h (km)
(d)

Kriging

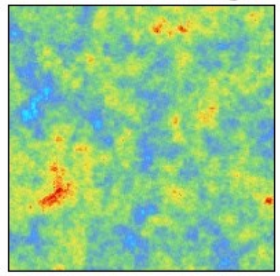
Rough



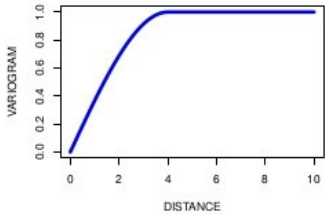
Smooth



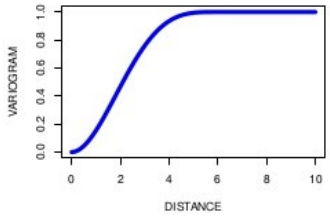
Rough



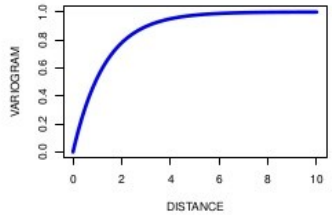
Spherical model



Cubic model

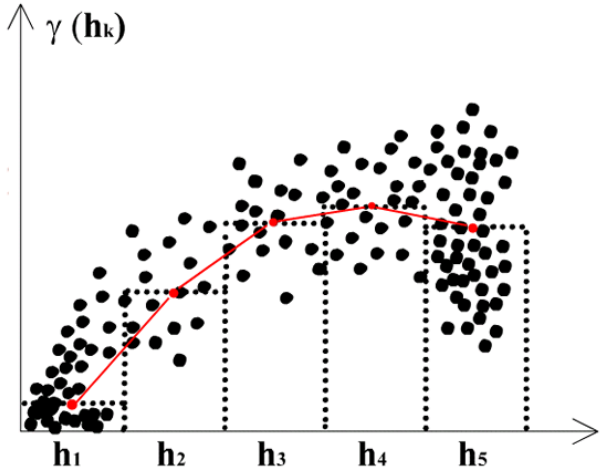


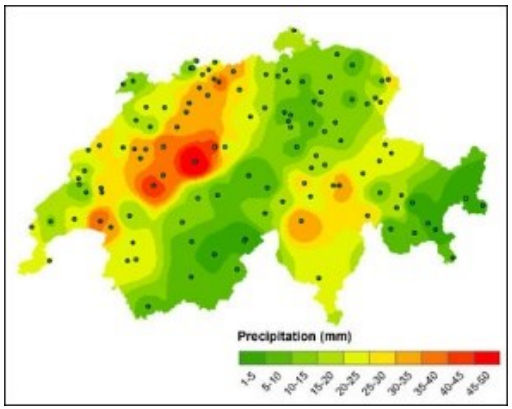
Exponential model



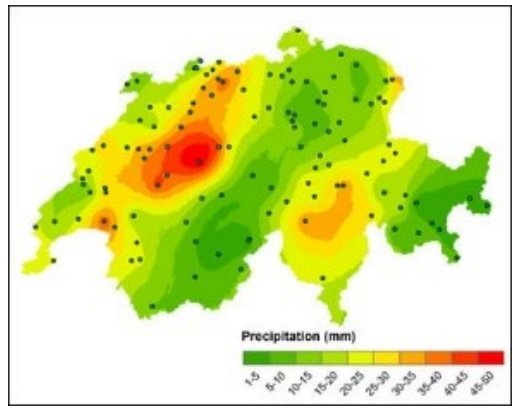
From a presentation by H. Wackernagel

Example IDW, Kriging





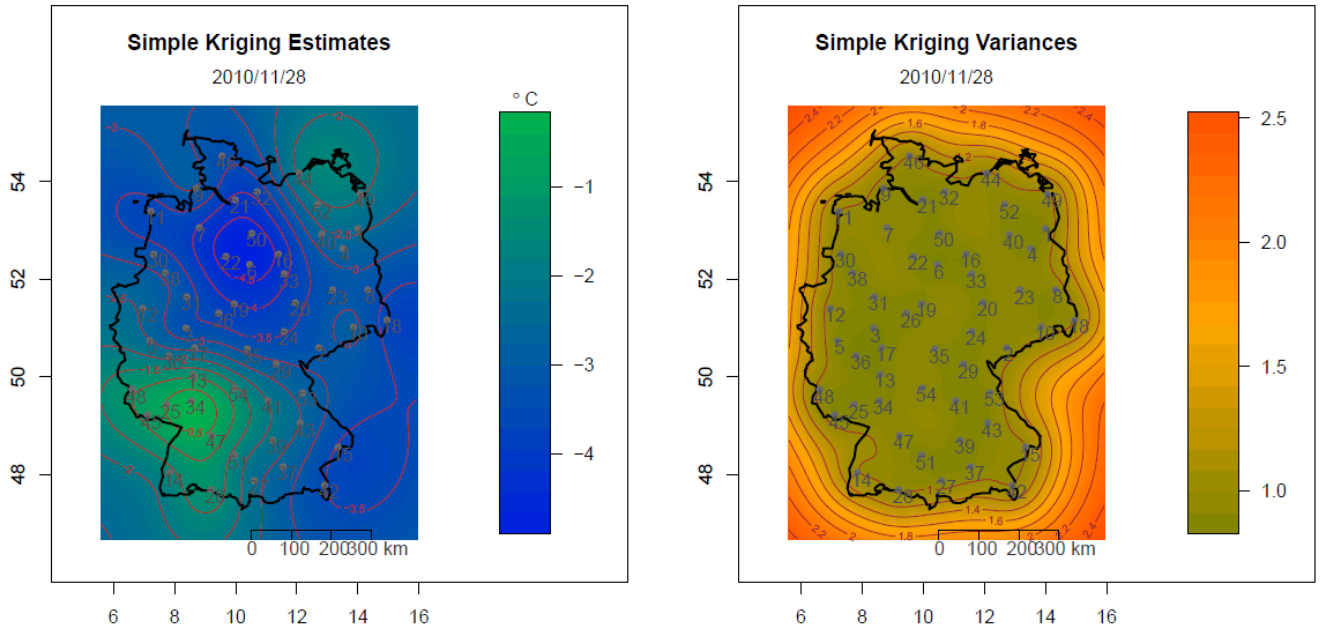
IDW



Kriging

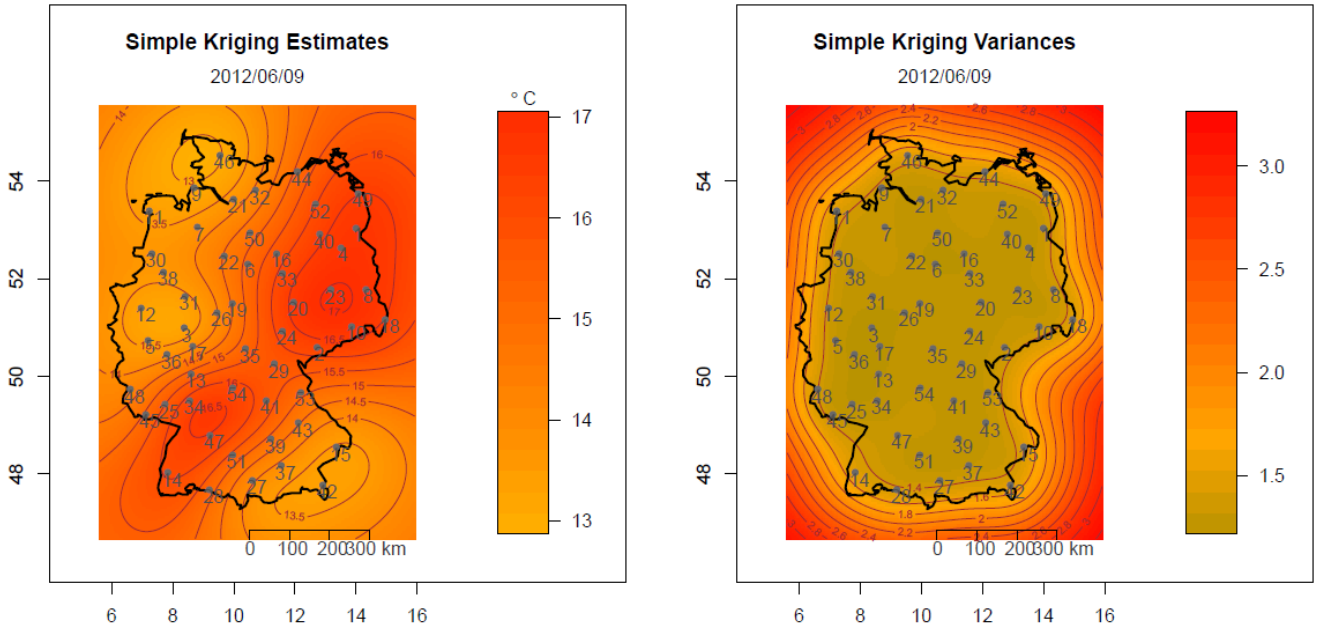
gitta.info

Example Kriging



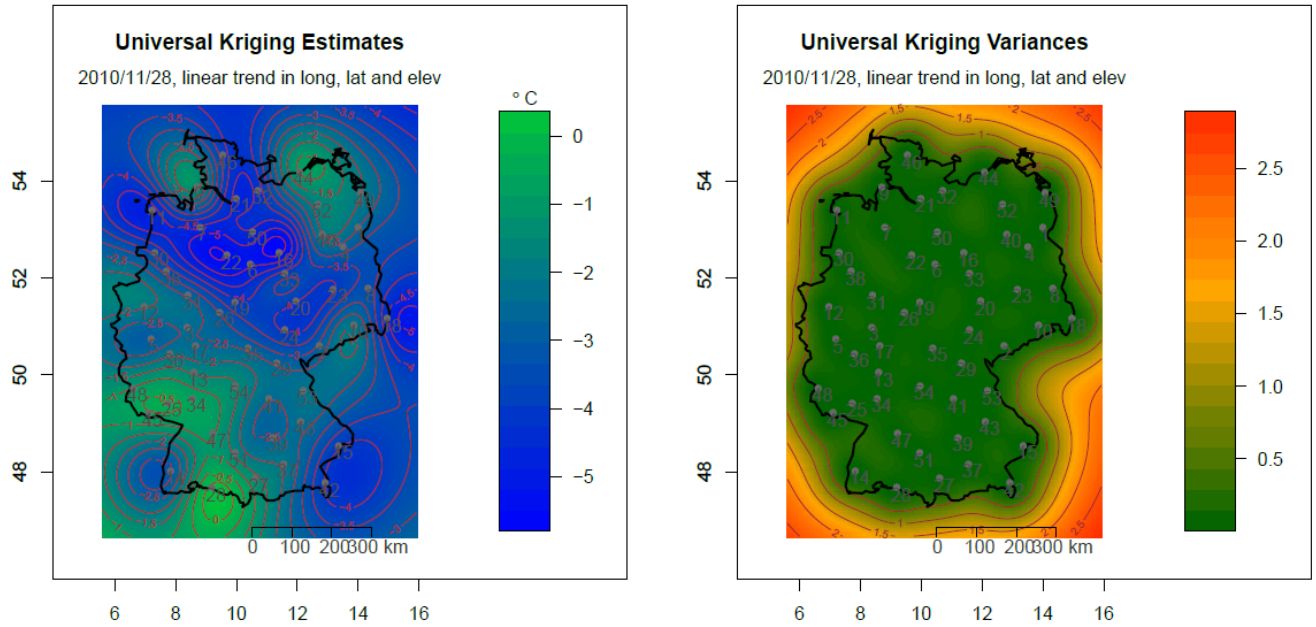
Example from the Bachelor thesis ”Kriging methods in spatial statistics”, Andreas Lichtenstern, TU München

Example Kriging



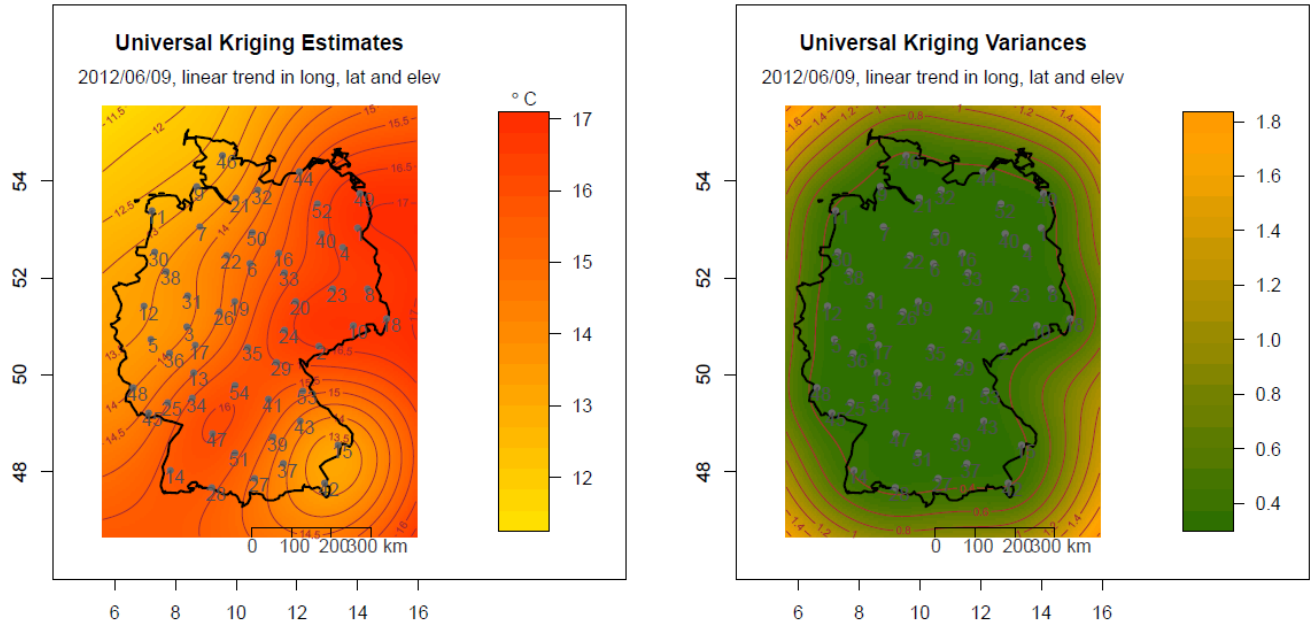
Example from the Bachelor thesis ”Kriging methods in spatial statistics”, Andreas Lichtenstern, TU München

Example Kriging



Example from the Bachelor thesis ”Kriging methods in spatial statistics”, Andreas Lichtenstern, TU München

Example Kriging



Example from the Bachelor thesis ”Kriging methods in spatial statistics”, Andreas Lichtenstern, TU München

Zusammenfassung: räumliche Vorhersage / Interpolation

Ausgangslage:
Gegeben sind bekannte Werte y_i (z.B. Messungen) für einen Parameter an gegeben Positionen x_i. An beliebigen Positionen x sollen die ggf. unbekannten Werte y vorhergesagt werden.

Interpolation:
Interpolation umfasst alle mathematischen Verfahren, welche einen Funktion $f(x) = y$ für eine solche Vorhersage verwenden. Klassischerweise hat bei einer linearen Interpolation $f(x)$ die Form eines gewichteten Mittels der gegeben Werte y_i:

$$f(x) = \sum_i \lambda_i(x) \cdot y_i$$

mit den Gewichten $\lambda_i(x)$. Die Bestimmung der Gewichte hängt vom gewählten Verfahren ab.

Approximation und Interpolation:
Für eine klassische (exakte) Interpolation gilt, dass das Verfahren die Daten exakt reproduziert mit

$$f(x_i) = y_i$$

Bei einer Approximation (oder auch nicht-exakte Interpolation) ist dies nicht der Fall, die Daten werden hier nur angenähert mit

$$f(x_i) = y_i + \epsilon_i \rightarrow f(x_i) \approx y_i$$

ϵ_i ist dabei der unbekannte Approximationsfehler.

Zusammenfassung: räumliche Vorhersage / Interpolation

Interpolation und Extrapolation:
Findet die Vorhersage innerhalb der Daten statt, handelt es sich um eine Interpolation. Die Ergebnisse können zumeist als plausibel angesehen werden, da genügend Daten als Randbedingungen für die Vorhersage zu Verfügung stehen.
Findet die Vorhersage außerhalb der Daten oder sehr weit entfernt statt, handelt es sich um eine Extrapolation. Die Ergebnisse können zumeist nicht als plausibel angesehen werden und sollte sehr kritisch interpretiert werden, da die Daten nur sehr eingeschränkt als Randbedingungen verwendet werden können.

Lokale und globale Interpolation:
Bei einer lokalen Interpolationen werden nicht alle Datenwerte für die Interpolation verwendet. Dies bedeutet, dass es Datenwerte y_i gibt, für das Interpolationsgewicht gleich Null ist:

$$\exists i: \lambda_i(x) = 0$$

Bei einer globalen Interpolationen werden immer alle gegebenen Werte berücksichtigt, d.h. dass es keine Interpolationsgewichte gibt, welche Null sind:

$$\forall i: \lambda_i(x) \neq 0$$

Gitterfreie und gitterbasierte Interpolation:
Benötigt ein Interpolationsverfahren eine Vermaschung der Datenpunkte wird es als *gitterbasiert* bezeichnet. Es ist dann zumeist ein lokales Verfahren, welches nur die Datenwerte verwendet, welche sich auf die Zelle beziehen, in der der Interpolationspunkt x liegt.
Gitterfreie (grid-free) Verfahren benötigen keine Vermaschung, sind aber zumeist globale Verfahren.

Zusammenfassung: räumliche Vorhersage / Interpolation

Deterministische Interpolation:
Die Bestimmung der Gewichte basiert auf einen festgelegten mathematischen Modell. Es ist zumeist nicht initial möglich, die „Unsicherheit“ der Vorhersage abzuschätzen.

(Geo-)statistische Interpolation (z.B. kriging):
Die Bestimmung der Gewichte basiert zusätzlich auf der bekannten oder abgeschätzten räumlichen Korrelation der Daten. Dies ist zumeist aufwändiger als deterministische Vorhersage, erlaubt aber die zusätzliche Abschätzung der „Unsicherheit“ der Vorhersage.

Wahl des Interpolationsverfahrens:
Bei der Wahl des Vorhersageverfahrens sollten folgende Fragen berücksichtigt werden:

1. Entspricht das mathematische Modell des Verfahrens dem vorherzusagenden Parameter bzgl.
   - Skale (diskret/kontinuierlich/…)
   - Glattheit / Differenzierbarkeit …
   - … ?
2. Ist Extrapolation für Anwendung notwendig?
3. Ist für die Anwendung exakte Interpolation notwendig oder reicht auch Approximation?
4. Welche Komplexität des Verfahrens ist für die Daten/Anwendung angemessen?
5. …

Übersicht

1. Definitionen, Funktionen, Anwendungen
2. Koordinatensysteme und -transformationen
3. Räumliche Datenmodellierung
4. Vermaschungen
5. Räumliche Interpolation
6. **Transformationen, Filtermethoden**

Filter Methoden

Anstelle eines „wahren“ Rasterbildes liegt typischerweise ein **verrauschtes Bild** vor, mit

$$f = f_{\text{true}} + n,$$

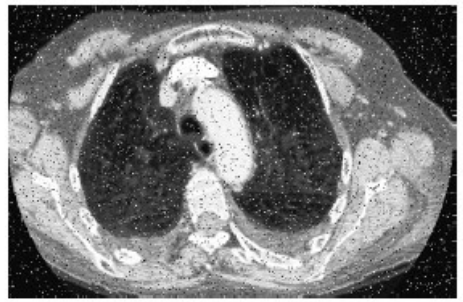
wobei der Fehler n mit verschiedenen Modellen beschrieben werden kann (z.B. Gauß-verteilt, „*gaussian noise*“). **Rauschentfernung / „*Denoising*“** beschreibt das Problem, eine Schätzung $g \approx f_{\text{true}}$ aus dem Wissen von f zu erhalten.



Original



Origial mit “*Gaussian noise*”



Origial mit “*Salt and Pepper noise*”

Filter Methoden

Anstelle eines „wahren“ Rasterbildes liegt typischerweise ein **verrauschtes Bild** vor, mit

$$f = f_{\text{true}} + n,$$

wobei der Fehler n mit verschiedenen Modellen beschrieben werden kann (z.B. Gauß-verteilt, „*gaussian noise*“). **Rauschentfernung / „*Denoising*“** beschreibt das Problem, eine Schätzung $g \approx f_{\text{true}}$ aus dem Wissen von f zu erhalten.

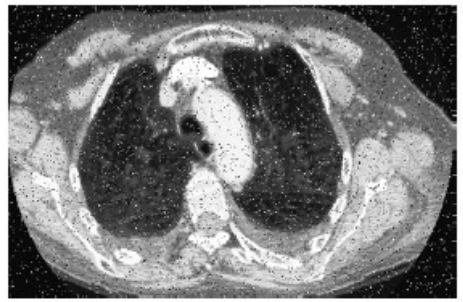
Rauschentfernung ist oft eine initialer Operation für weiter Bildbearbeitungsschritte.



Original



Original mit “*Gaussian noise*”



Original mit “*Salt and Pepper noise*”

Filter Methoden

Konvolution: Gegeben sein ein *Kernel* $w = [0,1]^2 \rightarrow \mathbb{R}$, die Konvolution einer Funktion f mit w ist definiert über

$$g(x,y) = (w * f)(x,y) = \int_{[0,1]^2} w(x-u, y-v) f(u,v) du, dv.$$

Fourier-Transformation: Die Fourier-Transformation einer Funktion f ist definiert über

$$F(u,v) = \hat{f}(u,v) = \int_{[0,1]^2} e^{-2\pi i (x,y)\cdot(u,v)} f(x,y) dx dy.$$

Konvolution und Fourier-Transformation: Die Fourier-Transformation einer Konvolution hat folgende Eigenschaft:

$$G(u,v) = W(u,v)F(u,v)$$

Filter Methoden

Konvolution: Gegeben sein ein *Kernel* $w = [0,1]^2 \rightarrow \mathbb{R}$, die Konvolution einer Funktion f mit w ist definiert über

$$g(x,y) = (w * f)(x,y) = \int_{[0,1]^2} w(x-u, y-v) f(u,v) du, dv.$$

Fourier-Transformation: Die Fourier-Transformation einer Funktion f ist definiert über

$$F(u,v) = \hat{f}(u,v) = \int_{[0,1]^2} e^{-2\pi i(x,y)\cdot(u,v)} f(x,y) dx dy.$$

Konvolution und Fourier-Transformation: Die Fourier-Transformation einer Konvolution hat folgende Eigenschaft:

$$G(u,v) = W(u,v)F(u,v)$$

Die Wahl der Kernelfunktion w oder W, entweder im Orts- oder Spektralbereich, bestimmt die Eigenschaften der Konvolution (Filter).

Filter Methoden

Diskrete Repräsentation / Pixel-Setup:

Konvolution: Gegeben sein ein Kernel $w \in \mathbb{R}^{(2K+1)\times(2L+1)}$, die diskrete Konvolution von f mit w ist definiert über

$$g(m,n) = (w * f)(m,n) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} w(k,l) f(m-k, n-l)$$

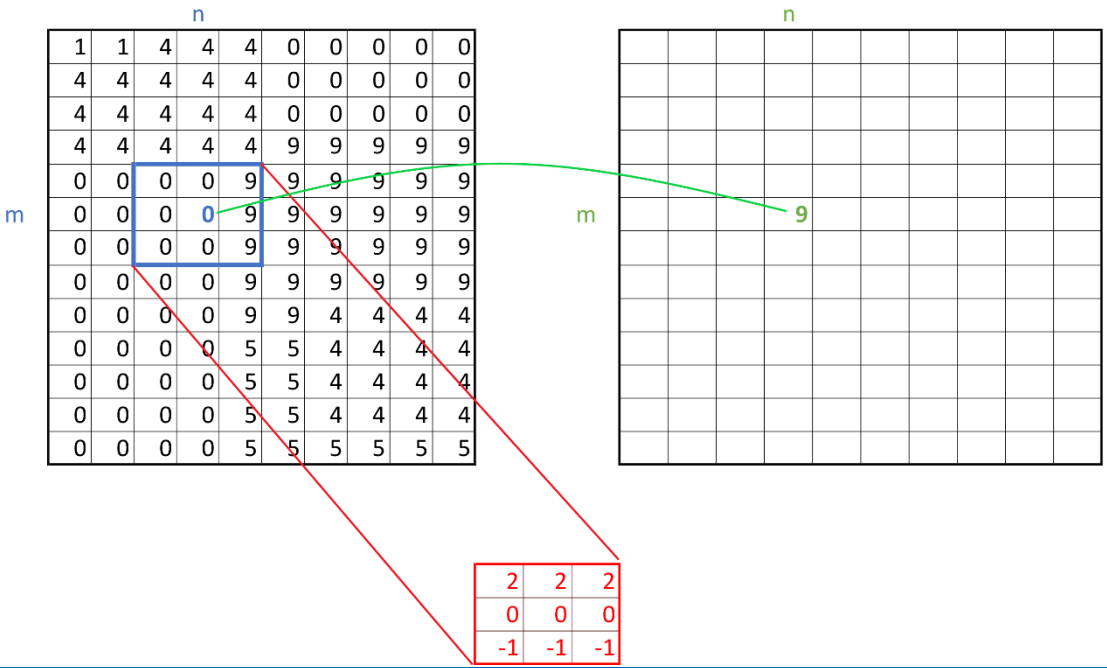
1	1	4	4	4	0	0	0	0	0
4	4	4	4	4	0	0	0	0	0
4	4	4	4	4	0	0	0	0	0
4	4	4	4	4	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	4	4	4	4
0	0	0	0	5	5	4	4	4	4
0	0	0	0	5	5	4	4	4	4
0	0	0	0	5	5	4	4	4	4
0	0	0	0	5	5	5	5	5	5

Filter Methoden

Diskrete Repräsentation / Pixel-Setup:

Konvolution: Gegeben sein ein Kernel $w \in \mathbb{R}^{(2K+1)\times(2L+1)}$, die diskrete Konvolution von f mit w ist definiert über

$$g(m,n) = (w * f)(m,n) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} w(k,l) f(m-k, n-l)$$



Filter Methoden

Beispiele für diskrete Kernelfunktionen w:

Mittelwertkernel / Averaging Kernel in einer $(2K+1) \times (2L+1)$-Nachbarschaft.
Für $L = K = 1$:

$$w = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Gauß-Kernel mit Gewichtsfunktion $w(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{\sigma^2}}$. Für $L = K = 2$ und $\sigma = 1$:

$$w = \frac{1}{271}\begin{pmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{pmatrix}.$$

Der Gaußfilter ist ein Spezialfall eines Diffusionsfilters.

0	0.2	0
0.2	0.2	0.2
0	0.2	0

5	3	6	1	5
4	4	7	4	5
6	5	2	1	2
7	6	1	0	2
8	6	2	1	3

Animation siehe OPAL! (https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975)

Filter Methoden

Original



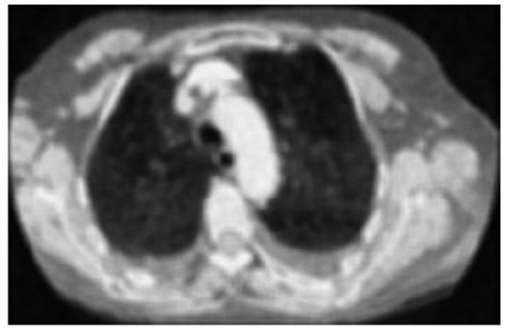
Gauß-verteiltes rauschen



Ergebnis Gaußfilter



Ergebnis Mittelwertfilter



Filter Methoden

Beispiele für nicht-lineare Filter (basieren nicht auf Konvolution):

Median Filter: Für jeden Pixel (m,n) wird eine lokale Umgebung $LA(m,n) = \{m-K,\dots,m+K\} \times \{n-L,\dots n+L\}$ definiert und es wird der Median aller Werte in der Umgebung zugeordnet:

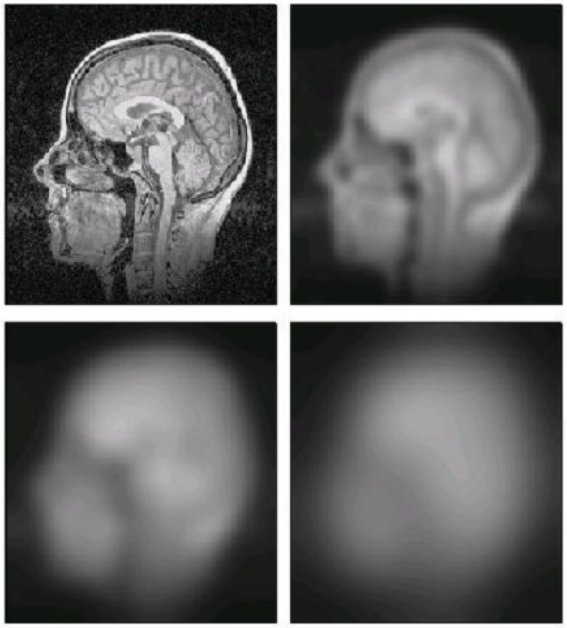
$$g(m,n) = \text{median}\{f(k,l): (k,l) \in LA(m,n)\}.$$

Nicht-lineare Diffusion: Sei D der Diffusions-Koeffizient, der von g oder ∇g (letztere Ausdruck ist mit Kanten assoziiert) abhängen kann. Dann ist die Diffusionsgleich gegeben mit

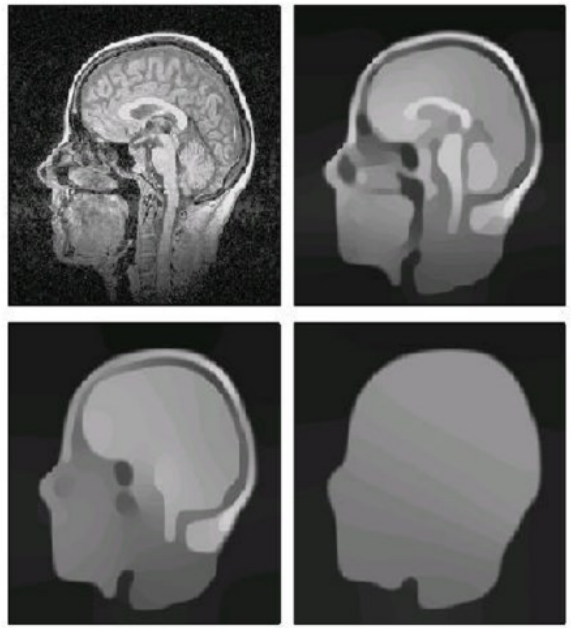
$$\frac{d}{dt} g(t,x,y) = \nabla \cdot \big(D(g,\nabla g)\big)\nabla g(t,x,y)$$
$$g(0,x,y) = f(x,y).$$

Oft wird $D = D(|\nabla g|) = e^{-\frac{|\nabla g|^2}{2\sigma^2}}$ gewählt (Perona-Malik Diffusion).

Filter Methoden



Scale space behaviour of **linear** Diffusion filtering



Scale space behaviour of **Nonlinear** Diffusion filtering

'Anisotropic Diffusion in Image Processing', J. Weickert, Teubner, 1998

Filter Methoden

Für welche Anwendung könnte man nicht-lineare Diffusionsfilterung noch verwenden?

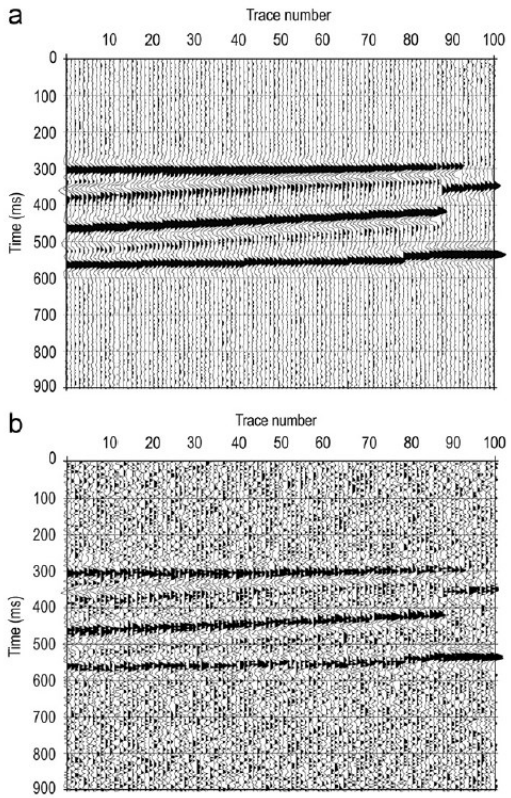


Fig. 3. Synthetic section contaminated with 10% (a) and 50% (b) random noise.

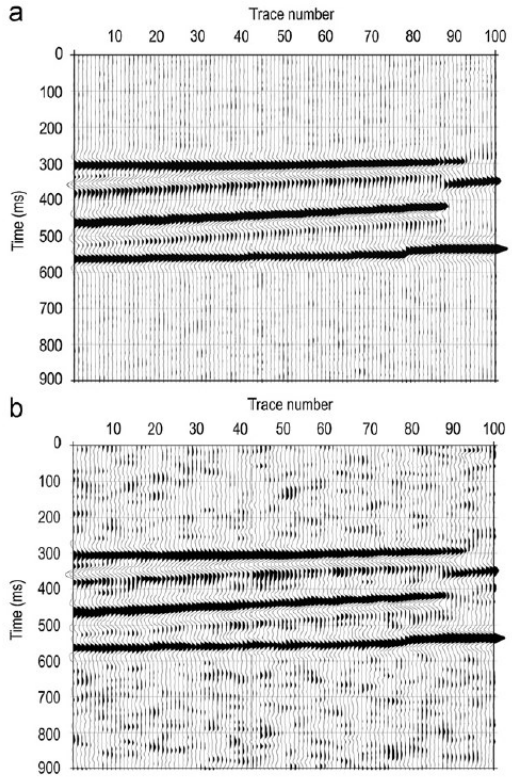
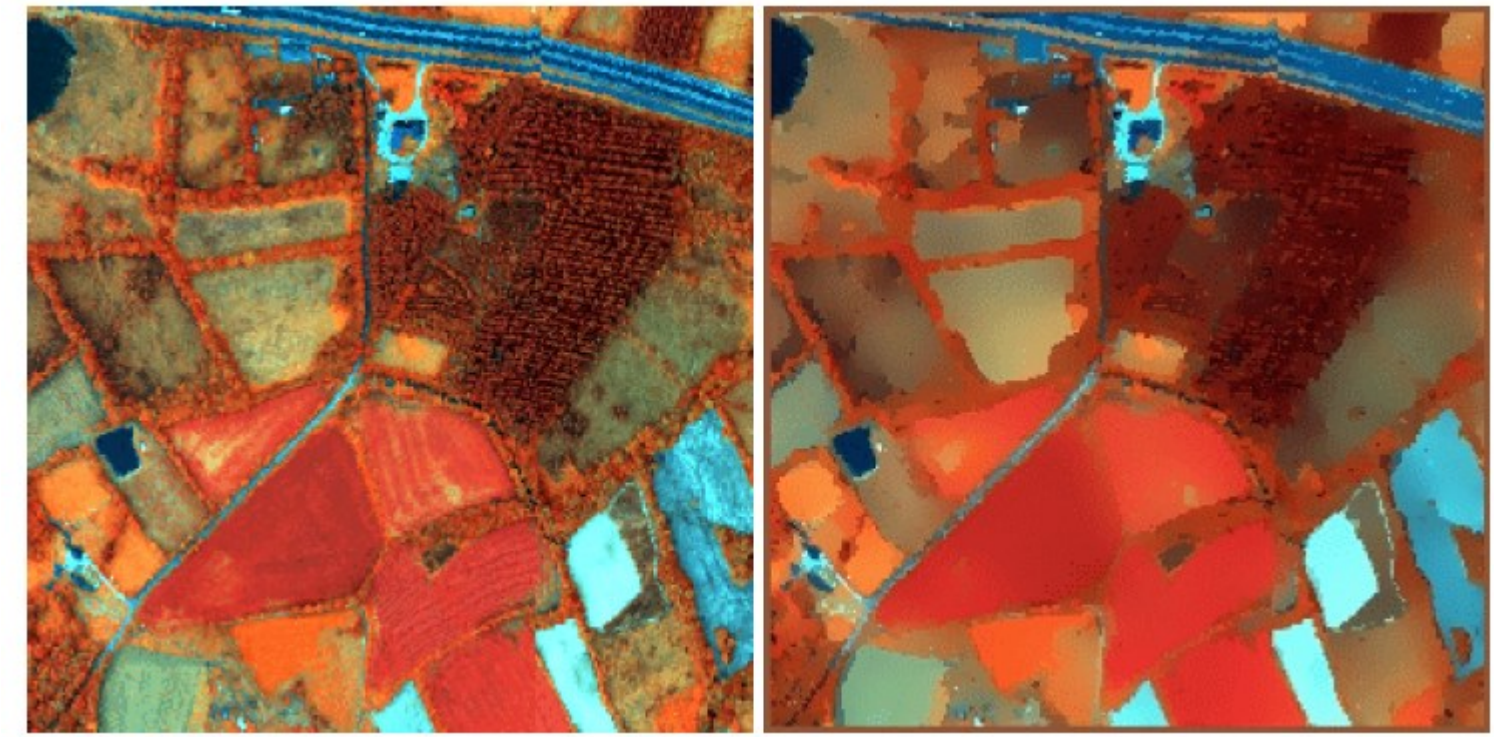


Fig. 4. Anisotropic non-linear diffusion filtering results: (a) section contaminated with 10% random noise and (b) section contaminated with 50% noise.

'Seismic noise attenuation by means of an anisotropic non-linear diffusion filter ', Baddari et al., Computers & Geoscience, 2011

Filter Methoden

Für welche Anwendung könnte man nicht-lineare Diffusionsfilterung noch verwenden?



‘Nonlinear filtering of hyperspectral images with anisotropic diffusion’, Lennon et al., IEEE Int. Geosc. Remote Sensing Symp., 2002

Image Segmentation

„*Image segmentation*“ bezieht sich auf eine große Gruppe von Anwendungen. Dabei kann es um die Unterteilung von Bildern in Vorder- und Hintergrund, die Unterteilung eines Bildes in Regionen ähnlicher Eigenschaften (z.B. Klassifikation) oder die Unterteilung in Regionen anhand sicherer Strukturen (z.B. Kanten).

Zwei Hauptkriterien zur Unterteilung sind:

1. **Diskontinuitäten**: Unterteilung anhand von Kanten

2. **Ähnlichkeit**: Unterteilung von ähnlichen Eigenschaften

Grundsätzlich gibt es keine universelle Unterteilungsmethode für alle Problemstellungen. Manuelle Unterteilung führt zwar häufig zu den „besten“ Ergebnissen, ist aber nur für sehr kleine Datensätze sinnvoll ist.

Kantendetektion / *Edge Detection*

Die Visualisierung / Detektion von kanten in Rasterbildern erfolgt typischerweise basierend auf Gradienten. Häufig werden dafür konvolutionelle Filter verwendet, wie z.B.:

Prewitt-Filter: Die folgenden Filterkernels werden für die Ableitung in x- und y-Richtung gewählt

$$w_x = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \qquad w_y = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

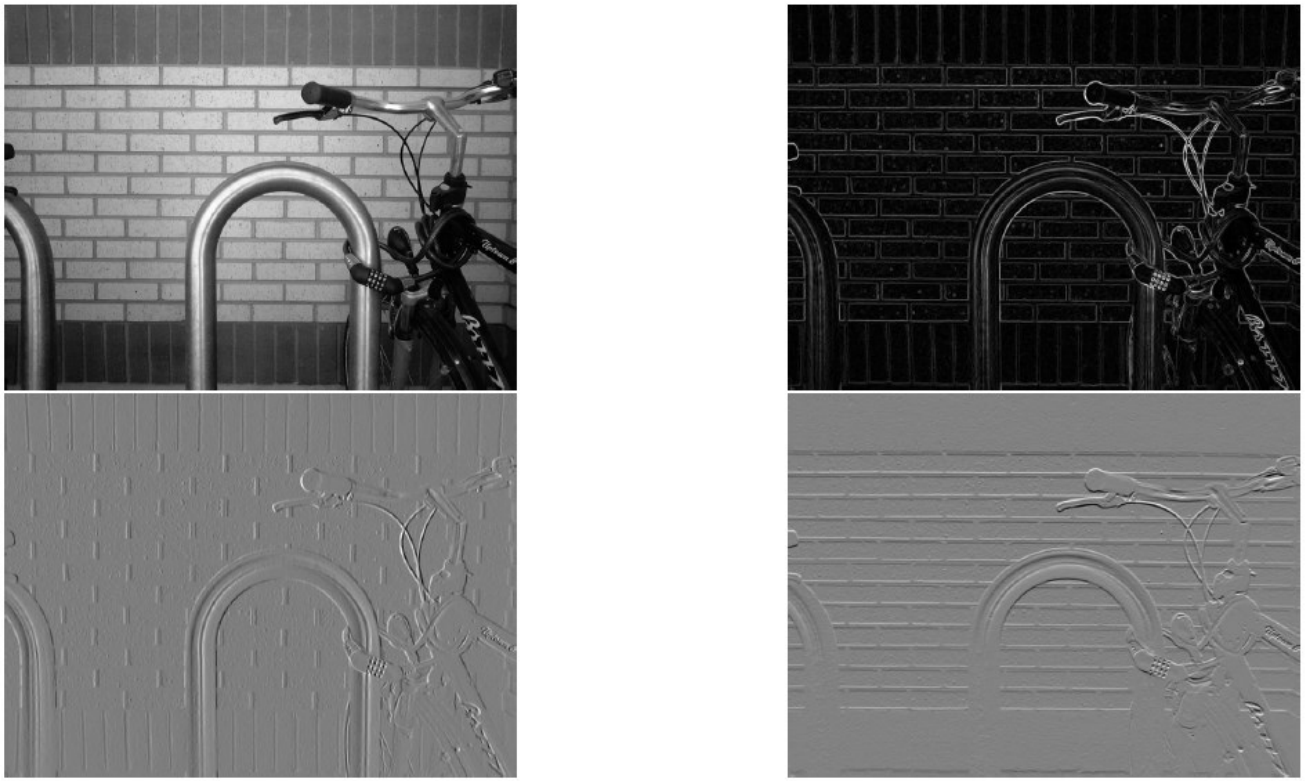
Sobel-Filter: Die folgenden Filterkernels werden für die Ableitung in x- und y-Richtung gewählt

$$w_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}, \qquad w_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

Kanten werden durch Pixel mit sehr hohem Gradienten repräsentiert. Alternativ kann man auch mit 2. Ableitung arbeiten. Diese ist genau dort Null, wo der Gradient hoch ist. Ein Verfahren, welches die die 2. Ableitung nutz ist, z.b. *Laplace zero crossing.*

Kantendetektion / *Edge Detection*

Oben links: Originalbild, unten links: Sobel-Filter in x-Richtung, unten rechts: Sobel-Filter in y-Richtung, oben rechts: Sobel-Filter kombiniert in x- und y-Richtung.



Ähnlichkeitssegmentierung über *Thresholding*

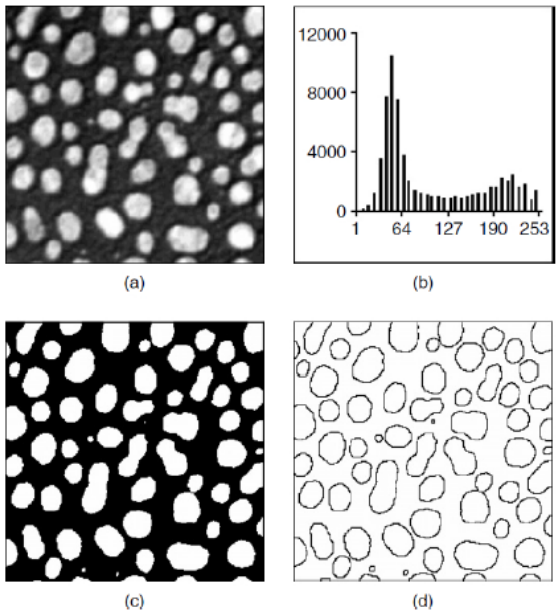
„*Thresholding*“ basiert auf der Ähnlichkeit von Pixelwerten eines Grauwert-Bildes. Besonders für Rasterdaten mit **bimodalem Histogramm** erreicht *Thresholding* gute Ergebnisse. Man wählt einen Intensitäts-Schwellwert (*threshold*) $T > 0$, das segmentierte Bild g ist definiert über

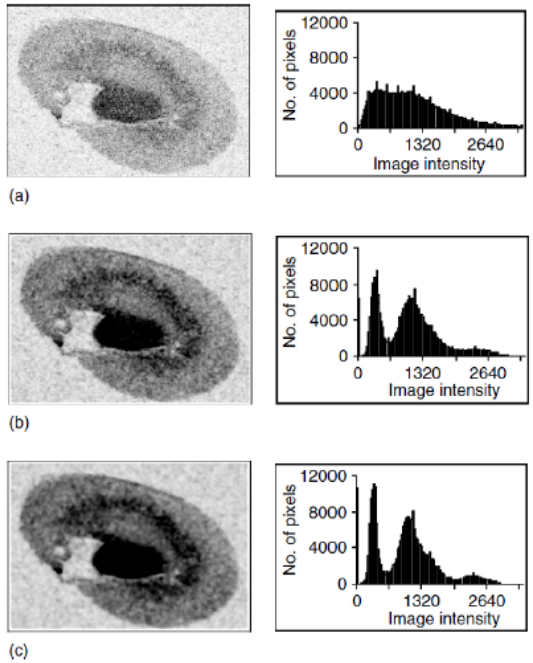
$$g(n,m) = \begin{cases} 1, & \text{if } f(n,m) > T, \\ 0, & \text{if } f(n,m) \leq T. \end{cases}$$

Alle Regionen mit $g(n,m) = 1$ können als „Vordergrund“ und alle anderen als „Hintergrund“ angesehen werden.

Ähnlichkeitssegmentierung über *Thresholding*

Verschiedene Segmentierungsverfahren lassen sich kombinieren.
Block links: (a) Original, (b) zugehöriges Historgramm, (c) Thresholding, (d) Laplace Zero Crossing; **Block rechts**: (a) Original + Histogramm, (b+c) Ergebnis + Histogramm nach Median-Filterung, (c) Filtern kann das Histogramm in ein bimodales Histogramm überführen, wodurch dann Thresholding angewendet werden kann.





Ähnlichkeitssegmentierung über *Thresholding*

Oft ist bei *Thresholding* initial nicht klar, welcher Schwellwert gewählt werden sollte. Eine automatisierte Methodik für dies ist z.B. die **Methode nach Otsu:**

1. Wähle initialen Schwellwert $T = 0$.

2. Unterteile f in Vordergrund g_1 and Hintergrund g_0

3. Berechne die Intra-Klassen-Varianz

$$\sigma(T)^2 = w_0(T)\sigma_0(T)^2 + w_1(T)\sigma_2(T)^2$$

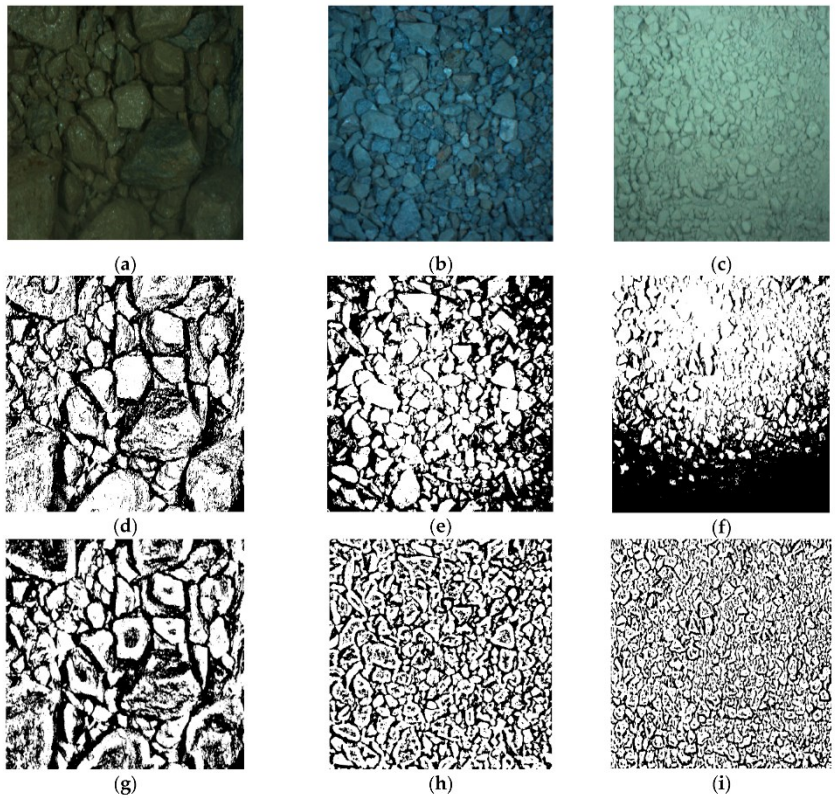
   mit w_0 ist der relative Anteil der Hintergrundpixel und w_1 ist der relative Anteil der Vordergrundpixel, $\sigma_0(T)^2$ ist die Varianz des Hintergrunds und $\sigma_1(T)^2$ ist die Varianz des Vordergrunds.

4. Iteriere über alle möglichen Schwellwerte T.

5. Wähle T mit der minimalen Intra-Klassen-Varianz.

Ähnlichkeitssegmentierung über *Thresholding*

Beispiel für Verfahren nach Otsu:
Oben: Original, Mitte: klassisches Verfahren nach Otsu, unten: verbesserte Variante von Otsu's Verfahren

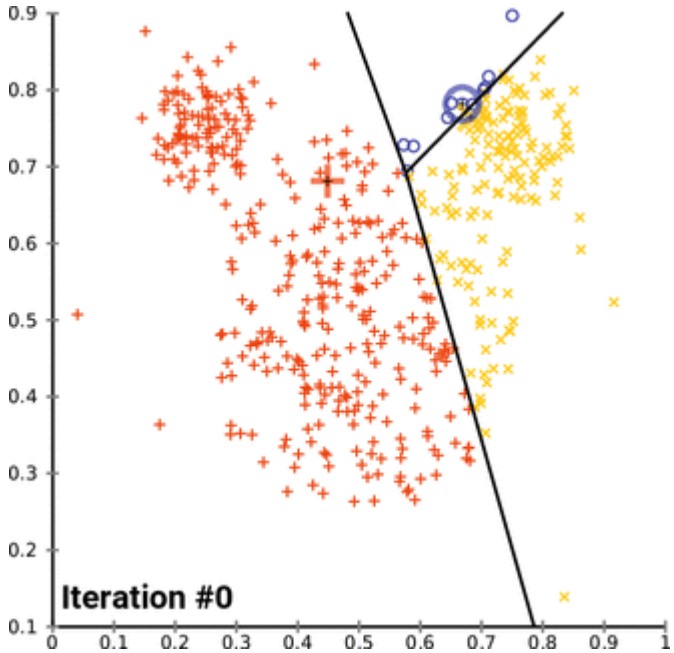


An Improved OTSU Algorithm using histogram accumulation moment for ore segmentation, Zhan and Zhang, Symmetry 11 (2019)

Ähnlichkeitssegmentierung über *kMeans-Klassifikation*

Gegen sein Datenpunkte $\mathbf{x}_i, i = 1, \dots, N$. Klassifikation diesen Datenpunkte basierend auf dem Abstand im Parameterraum zu Klassenzentren $\mu_k, k = 1, \dots, K$ für die Klassen S_k. Es gilt, folgende Zielfunktion zu minimieren:

$$\mathrm{J} = \sum_{k=1}^{K} \sum_{\mathbf{x}_j \in S_k} \left\| \mathbf{x}_j - \mu_k \right\|^2$$



Ähnlichkeitssegmentierung über *kMeans-Klassifikation*

Gegen sein Datenpunkte $\mathbf{x}_i, i = 1, \dots, N$. Klassifikation diesen Datenpunkte basierend auf dem Abstand im Parameterraum zu Klassenzentren $\mu_k, k = 1, \dots, K$ für die Klassen S_k. Es gilt, folgende Zielfunktion zu minimieren:

$$\mathrm{J} = \sum_{k=1}^{K} \sum_{\mathbf{x}_j \in S_k} \left\| \mathbf{x}_j - \mu_k \right\|^2.$$

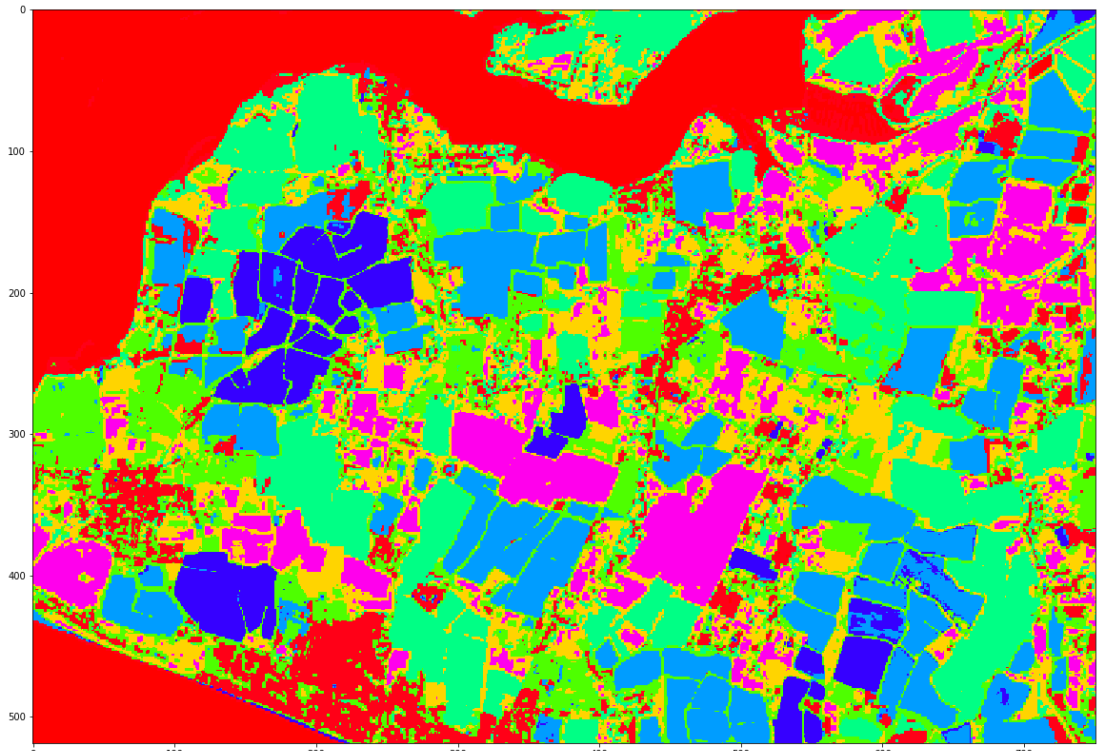
Im Allgemeinen wird versucht, jedem Datenpunkt $\mathbf{x}_i$der Klasse zuzuordnen, deren Varianz sich durch das Hinzufügen von $\mathbf{x}_j$ am geringsten erhöht.

Allg. iteratives Vorgehen:

1. Initialisiere K Klassen mit ihren Zentren $\mu_k^i, i = 0$.
2. $\forall \mathbf{x}$: Ordne $\mathbf{x}_i$ der Klasse zu, für die $\left\| \mathbf{x}_i - \mu_k^i \right\|^2$, minimal ist.
3. $\forall \mu_k^i$: Update die Position von μ_k: $\mu_k^{i+1} = \frac{\sum_{\mathbf{x}_j \in S_k^i} \mathbf{x}_j}{|\mathbf{x}_j \in S_k^i|}$
4. Wiederhole 2 – 4 mit $i = \mathrm{i} + 1$ bis Abbruchbedingung erfüllt, z.B. $\mu_k^i = \mu_k^{i+1}, \forall \mu_k^i$.

Ähnlichkeitssegmentierung über *kMeans-Klassifikation*

1. Nicht-überwachtes Klassifikationsverfahren (un-supervised).
2. Anzahl der Klassen steht fest.
3. Optimale Wahl der Metrik für $\left\|\mathbf{x}_i - \mu_k^i\right\|^2$ und der Abbruchbedingung für Ergebnis entscheidend, abhängig vom Problemstellung.
4. Sehr einfache Implementierung.



https://www.acgeospatial.co.uk/k-means-sentinel-2-python/

Morphologische Operationen

Morphologische Operationen sind auf den ersten Blick mit Filtern verwandt, aber eher mengentheoretisch orientiert. Sie dienen u.a. dazu geometrische Informationen in Bildern zu analysieren und transformieren. Sie können mit der Anwendung von Filtern kombiniert werden.

Die Grundoperationen sind **Erosion** und **Dilitation**. Wir betrachten Sie hier nur für binäre Bilder (mit Bild meinen wir hier eine Funktion $f : \{1, \ldots, N\} \times \{1, \ldots, M\} \to \{0, 1\}$).

Morphologische Operationen

Morphologische Operationen sind auf den ersten Blick mit Filtern verwandt, aber eher mengentheoretisch orientiert. Sie dienen u.a. dazu geometrische Informationen in Bildern zu analysieren und transformieren. Sie können mit der Anwendung von Filtern kombiniert werden.

Die Grundoperationen sind **Erosion** und **Dilitation**. Wir betrachten Sie hier nur für binäre Bilder (mit Bild meinen wir hier eine Funktion $f : \{1, \ldots, N\} \times \{1, \ldots, M\} \to \{0, 1\}$).

Die Translation einer Menge $X \subset \{1, \ldots, N\} \times \{1, \ldots, M\}$ ist definiert durch

$$X_q = \{p \in \{1, \ldots, N\} \times \{1, \ldots, M\} : p - q \in X\}$$

Ein translatiertes Bild ist demnach gegeben durch $f_q(p) = f(p - q)$.
Als Punktinversion bezeichnen wir

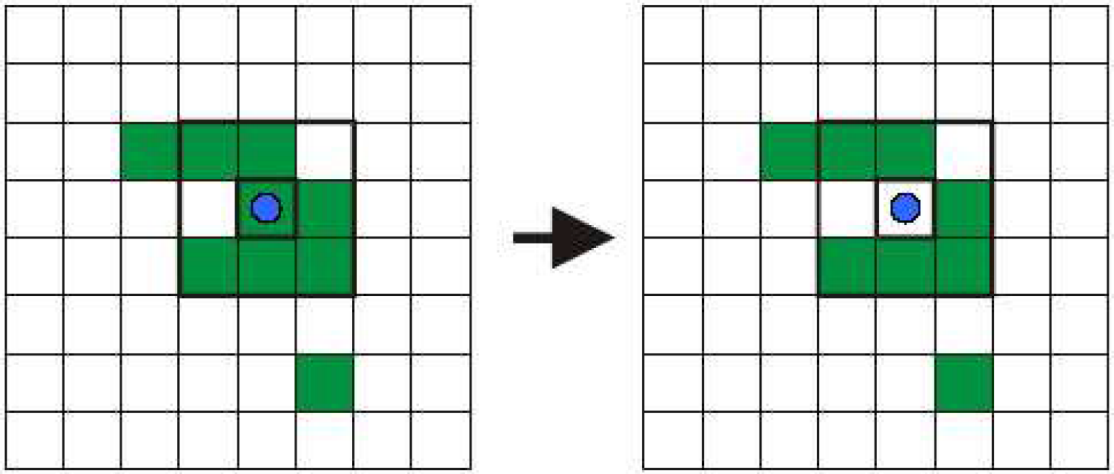
$$\check{X} = \{-p : p \in X\}$$

Morphologische Operationen

Die **Erosion** einer Menge X mit einem Strukturelement B ist nun definiert als

$$\varepsilon_B(X) = \{p \in \{1, \ldots, N\} \times \{1, \ldots, M\} : B_p \subset X\} = \bigcap_{p \in B} X_{-p}.$$

In Worten: Erosion beschreibt alle Pixel p, für die das Strukturelement B mit Zentrum p in X liegt.

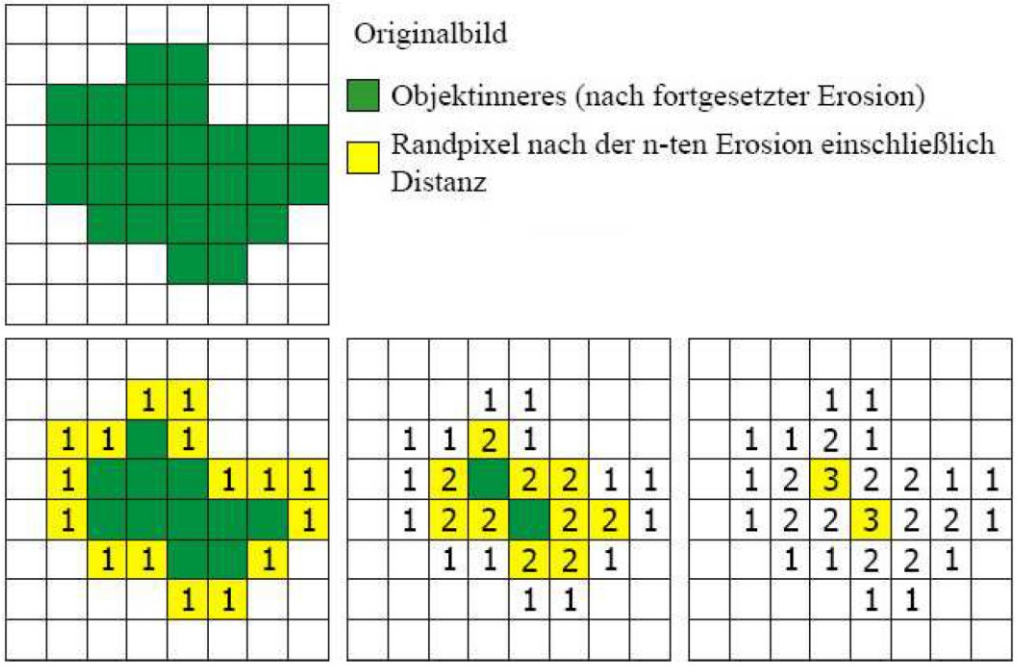


Morphologische Operationen

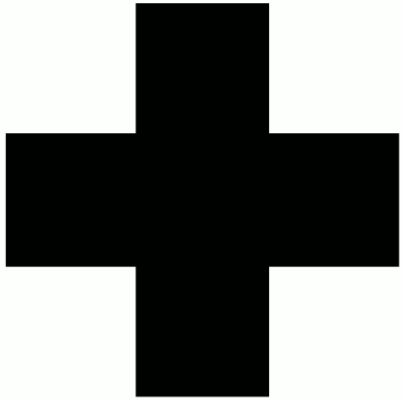
Die **Erosion** einer Menge X mit einem Strukturelement B ist nun definiert als

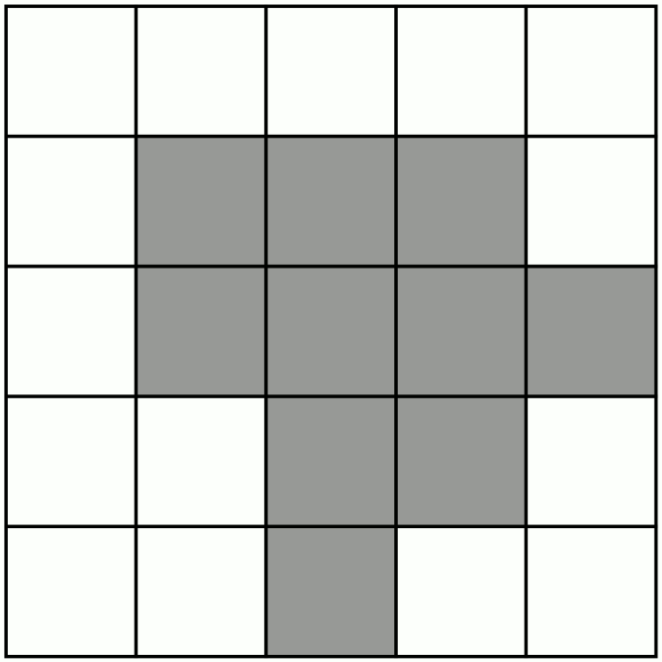
$$\varepsilon_B(X) = \{p \in \{1, \ldots, N\} \times \{1, \ldots, M\} : B_p \subset X\} = \bigcap_{p \in B} X_{-p}.$$

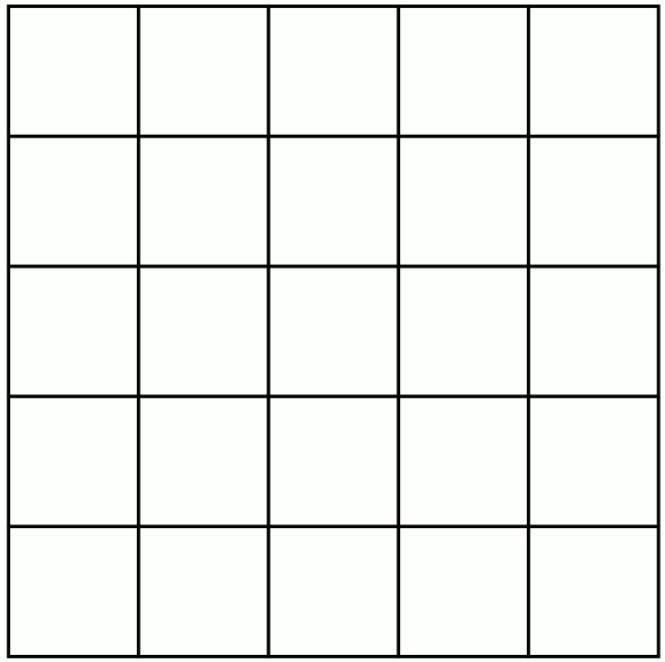
In Worten: Erosion beschreibt alle Pixel p, für die das Strukturelement B mit Zentrum p in X liegt.



Welches Strukturelement wurde hier gewählt?







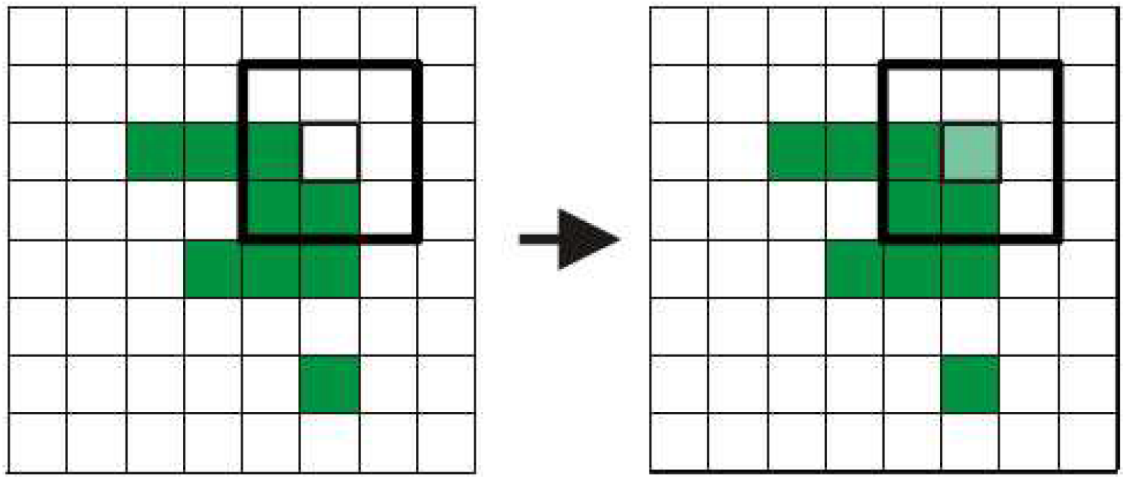
[Animation siehe OPAL!](https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975) (https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975)

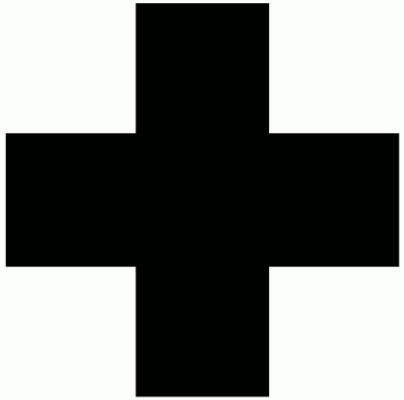
Morphologische Operationen

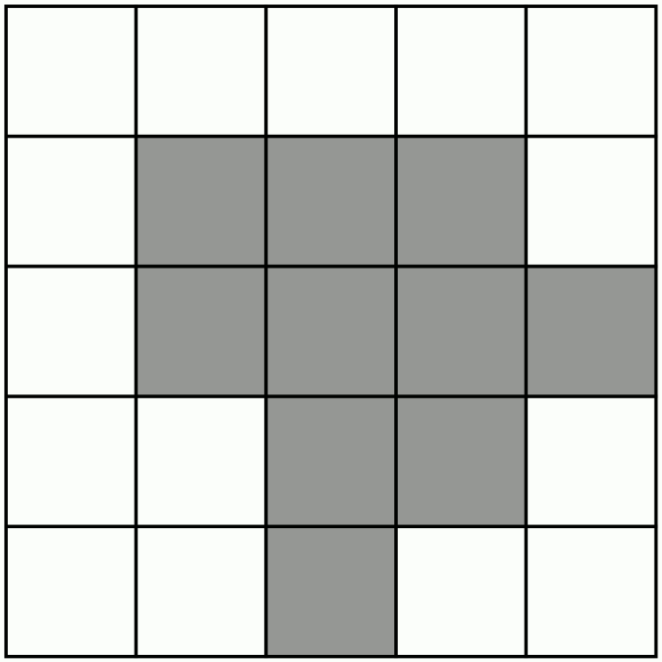
Die **Dilatation** einer Menge X mit einem Strukturelement B ist nun definiert als

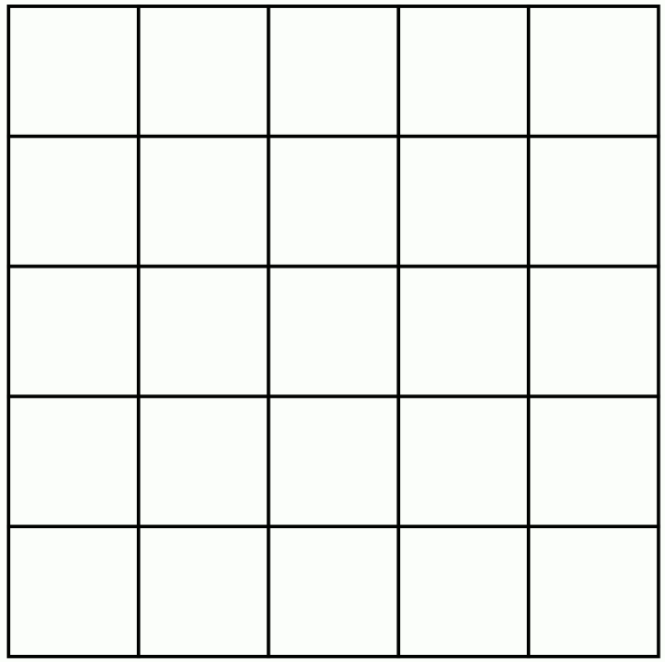
$$\delta_B(X) = \{p \in \{1, \ldots, N\} \times \{1, \ldots, M\} : B_p \cap X \neq \emptyset\} = \bigcup_{p \in B} X_{-p}.$$

In Worten: Dilatation beschreibt alle Pixel p, für die das Strukturelement B mit Zentrum p mindestens ein Element der Menge X enthält.









[Animation siehe OPAL!](https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975) (https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975)

Morphologische Operationen

Die **Öffnung** einer Menge X mit einem Strukturelement B ist definiert als

$$\gamma_B(X) = \{B_p : p \in \{1, \ldots, N\} \times \{1, \ldots, M\}, B_p \subset X\} = \delta_{\check{B}}(\varepsilon_B(X)).$$

In Worten: öffnung beschreibt beschreibt die Vereinigung aller Strukturelemente B_p, die in X enhalten sind, nicht nur der Referenzpunkte p (lezteres ist Erosion).

Morphologische Operationen

Die **Öffnung** einer Menge X mit einem Strukturelement B ist definiert als

$$\gamma_B(X) = \{B_p : p \in \{1, \ldots, N\} \times \{1, \ldots, M\}, B_p \subset X\} = \delta_{\check{B}}(\varepsilon_B(X)).$$

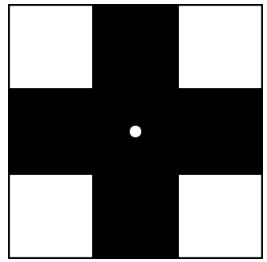
In Worten: öffnung beschreibt beschreibt die Vereinigung aller Strukturelemente B_p, die in X enhalten sind, nicht nur der Referenzpunkte p (lezteres ist Erosion).

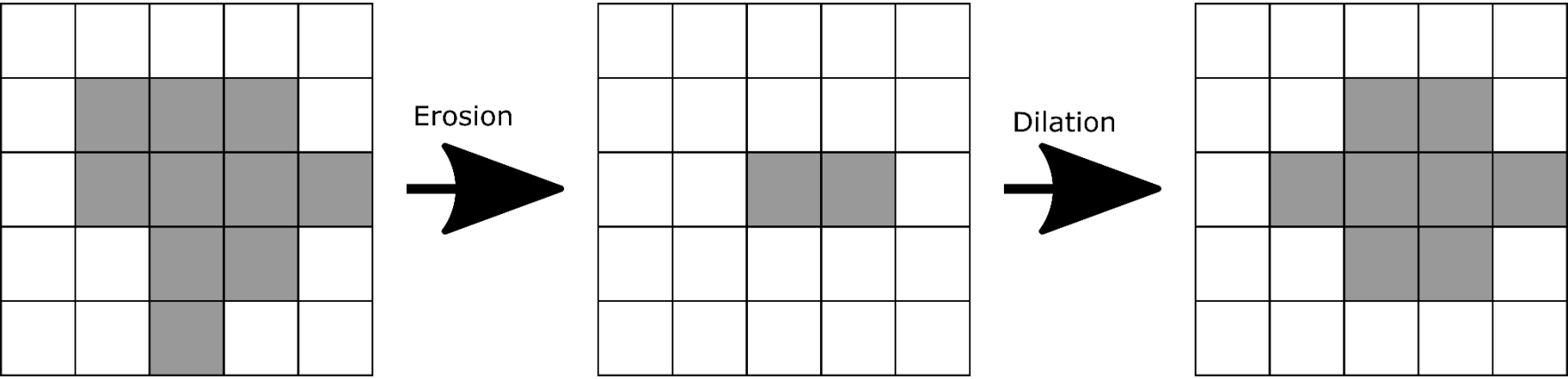
Die **Schliessung** einer Menge X mit einem Strukturelement B ist definiert als

$$\begin{aligned}\phi_B(X) &= (\{B_p : p \in \{1, \ldots, N\} \times \{1, \ldots, M\}, B_p \subset X^c\})^c \\ &= \bigcap_{p \in \{1,\ldots,N\} \times \{1,\ldots,M\}} \{B_p^c : X \in B_p^c\} = \varepsilon_{\check{B}}(\delta_B(X)).\end{aligned}$$

Morphologische Operationen

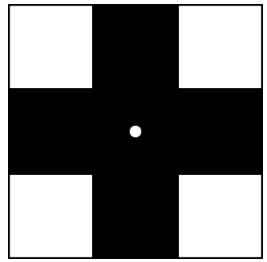
Öffnung / opening

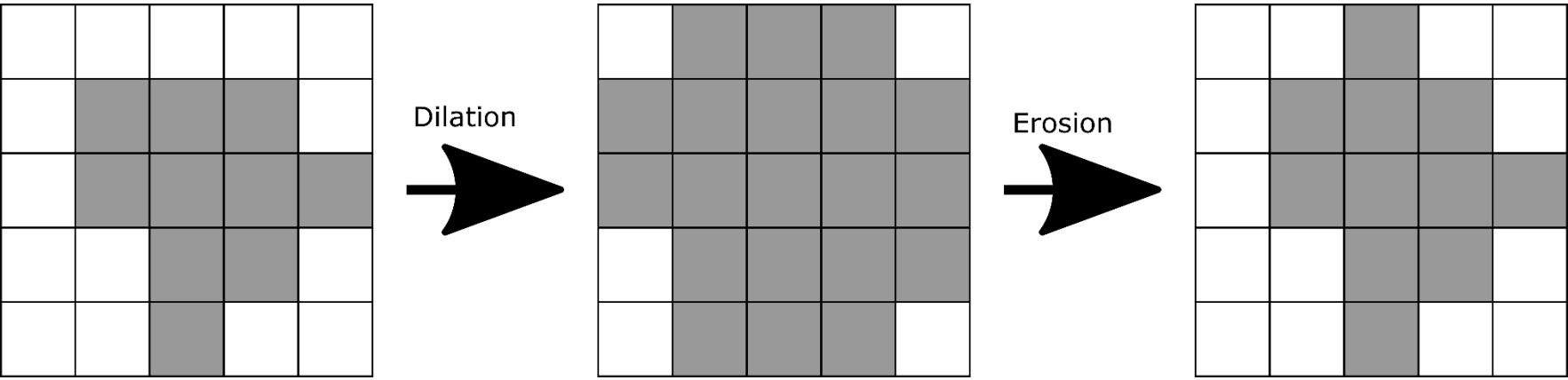




Morphologische Operationen

Schließung / Abschluss / closure





Morphologische Operationen

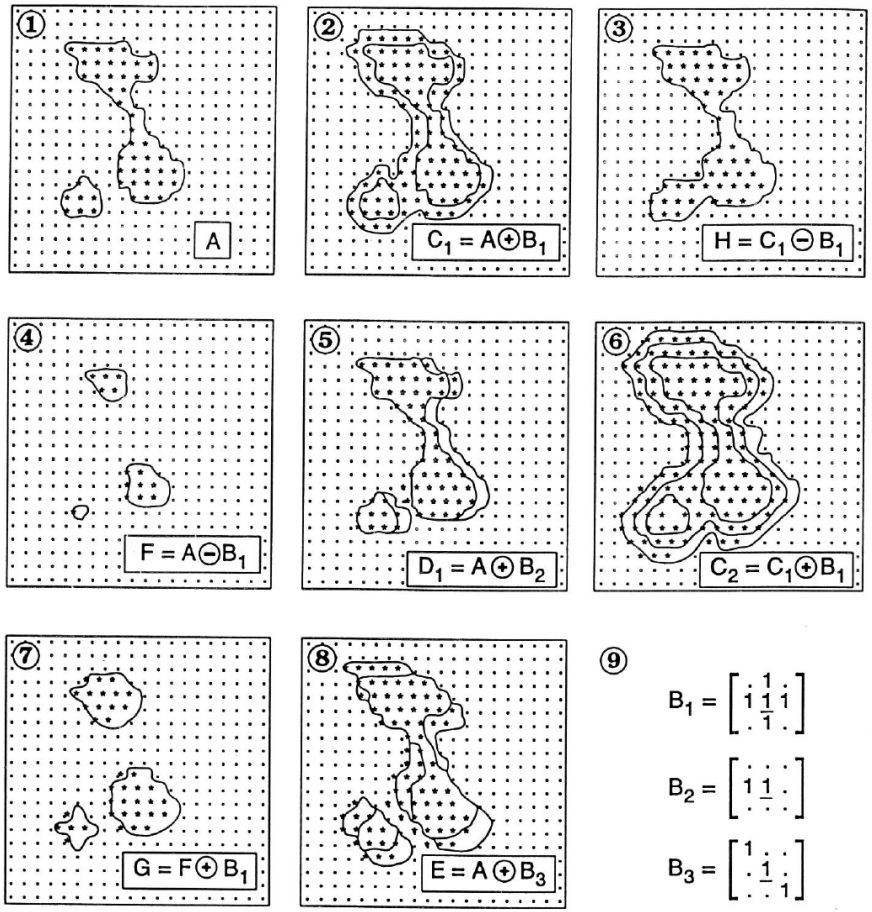


FIG. 6-9. Mathematical morphology operations on a small raster image. **(1).** Original image **A**. **(2).** Dilation of **A** using structuring element B_1. **(3).** Closing of image **A** by B_1. **(4).** Erosion of **A** by B_1. **(5).** Dilation of **A** by structuring element B_2. **(6).** Two successive dilations of **A** by B_1. **(7).** Opening of **A** by B_1. **(8).** Dilation of **A** by B_3. **(9).** Three (3*3) structuring elements.

Morphologische Operationen

Arithmetische Differenzen

$$\delta_B(f) - \varepsilon_B(f)$$
$$\delta_B(f) - f$$
$$f - \varepsilon_B(f)$$

können genutzt werden, um Ränder in Bildern hervorzuheben (ähnlich wie Sobel- oder Prewittfilter).

Morphologische Operationen

Arithmetische Differenzen

$$\delta_B(f) - \varepsilon_B(f)$$
$$\delta_B(f) - f$$
$$f - \varepsilon_B(f)$$

können genutzt werden, um Ränder in Bildern hervorzuheben (ähnlich wie Sobel- oder Prewittfilter).

Öffnungen und Schliessungen können genutzt werden, um bestimmte geometrische Objekte in Bildern herauszufiltern oder zu erzwingen.

Morphologische Operationen

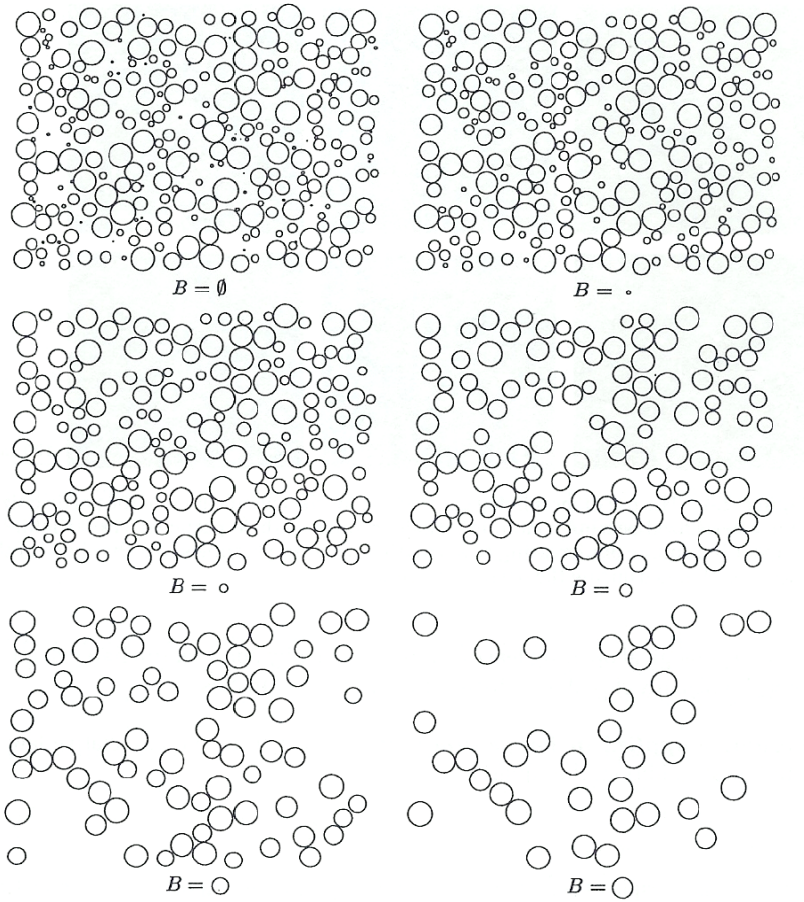


Figure 2.3 'Sieving' of disks of uniformly distributed diameter. The structuring element B is a disk of increasing diameter. If the diameter of the disk is smaller than the diameter of the current structuring element, this disk is eliminated from the image. For $B = \emptyset$ we have the original image.

Morphologische Operationen

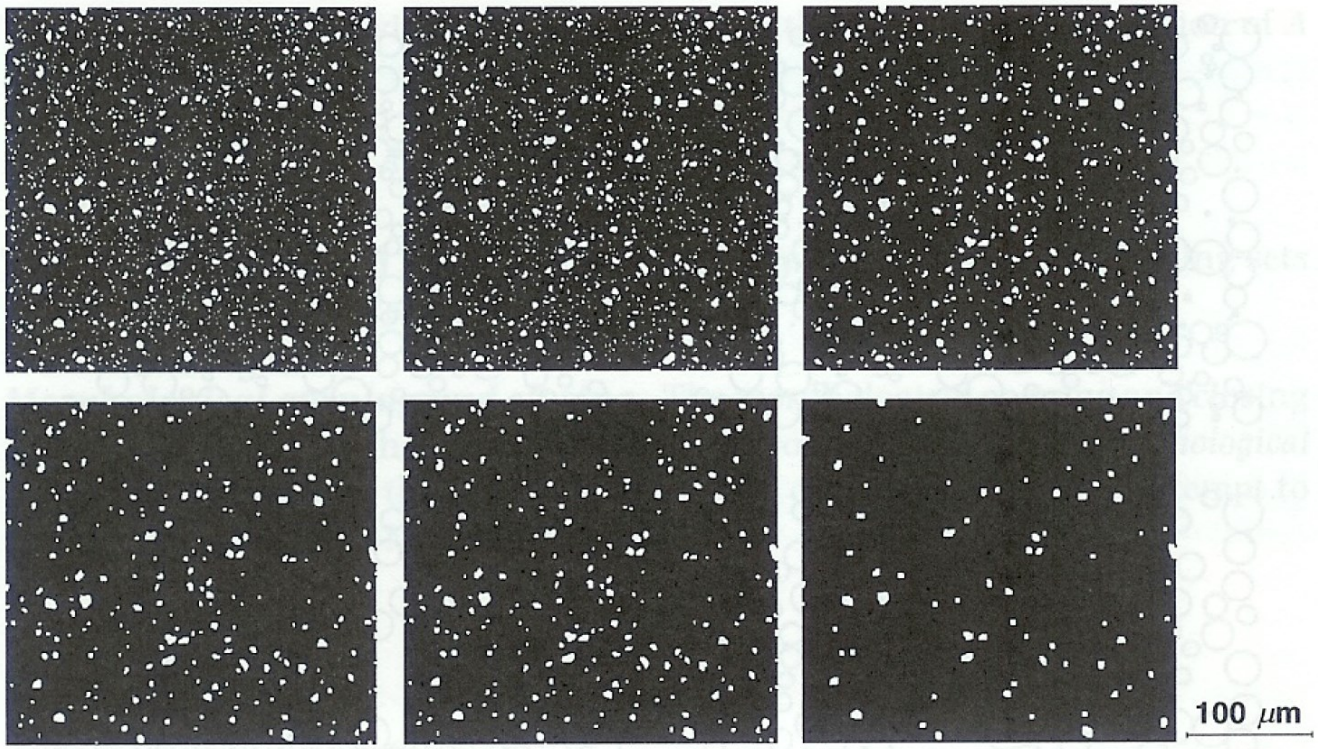
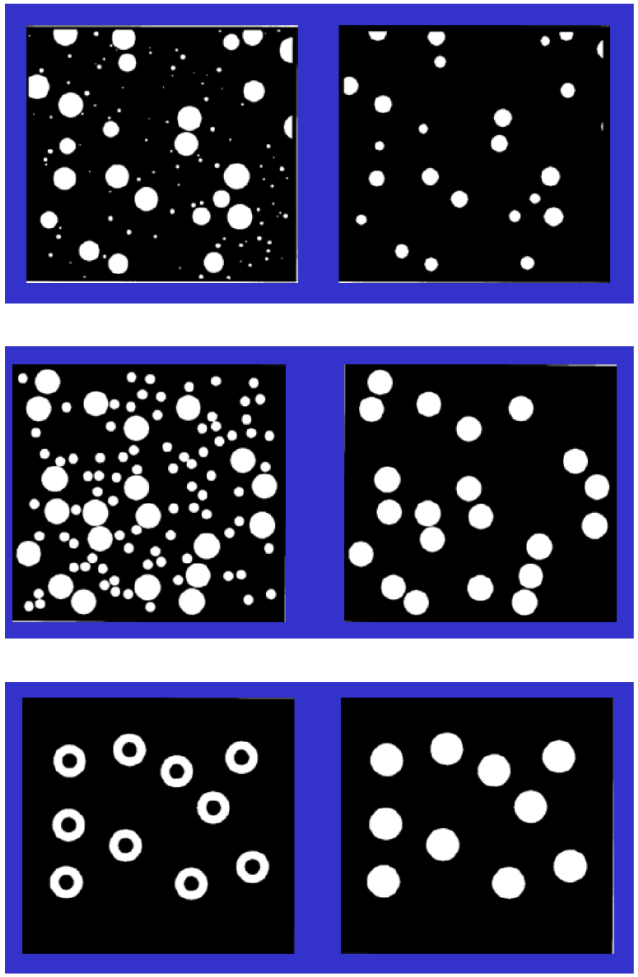


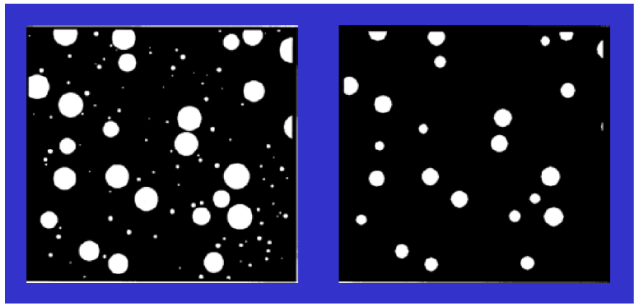
Figure 2.4 The use of morphological opening for the classification of the particle sizes of a spheroidized pearlite containing small granular carbide particles. The structuring element is a square of increasing edge length. This example shows that the shape of the objects is changed during the opening process.

Morphologische Operationen

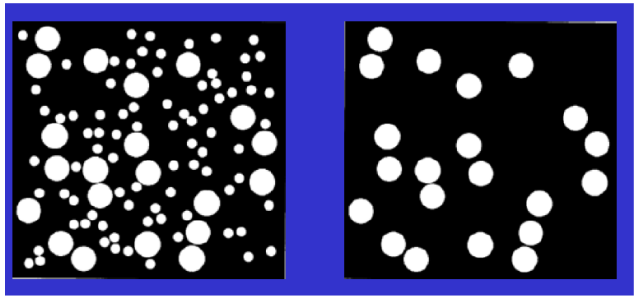


Welche Operationen wurden durchgeführt?

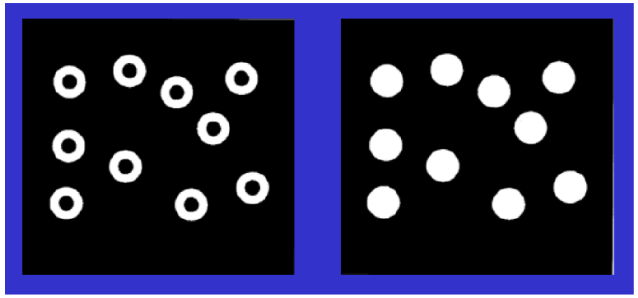
Morphologische Operationen



Erosion

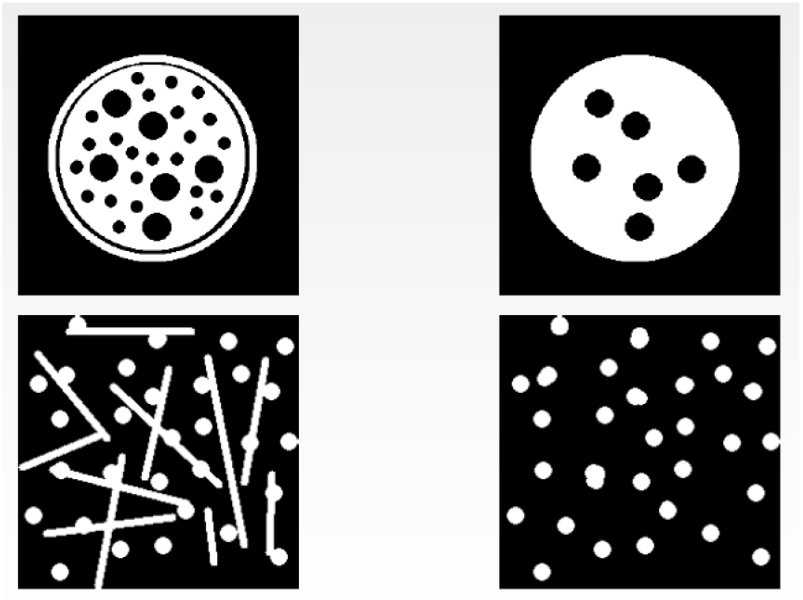


Öffnung



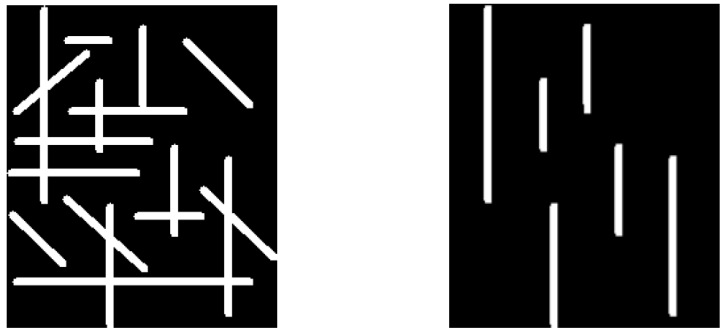
Schliessung

Morphologische Operationen

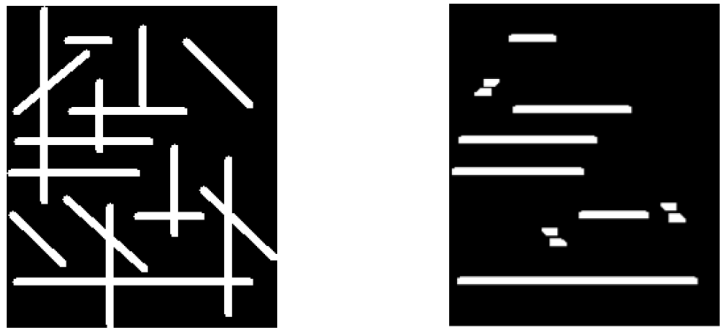


top: Closing with a disc shaped structuring element
bottom: Opening with a disk shaped structuring element

Morphologische Operationen



Opening with a 9×3 vertically oriented structuring element.



Opening with a 3×9 horizontally oriented structuring element.

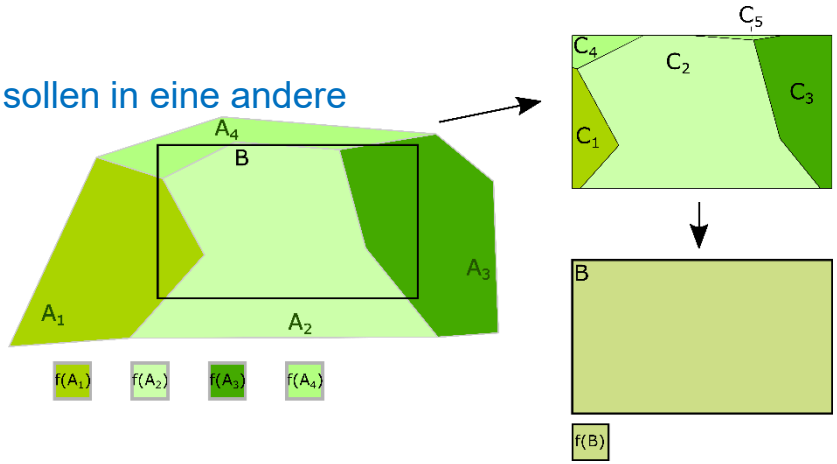
Sampling Transformationen

Probleme

- Gegeben seien mehrere verschiedene Geoobjekte. Wie lassen sich dieses anhand ihres relativen Abstands klassifizieren?
  - Buffering

- Gegeben sind ein oder mehrere Geoobjekte mit Attributen. Diese Attributwerte sollten jetzt für einen beliebigen Punkt oder eine Region abgefragt/ermittelt werden …
  - Punkt → Punkt: **Interpolation!**
  - Linie → Punkt: **Interpolation!**
  - Fläche → Punkt …
  - Fläche → Fläche …

- Geoobjekte in einer gegebenen Auflösung sollen in eine andere Auflösung überführt werden (*Resampling*)



Buffering

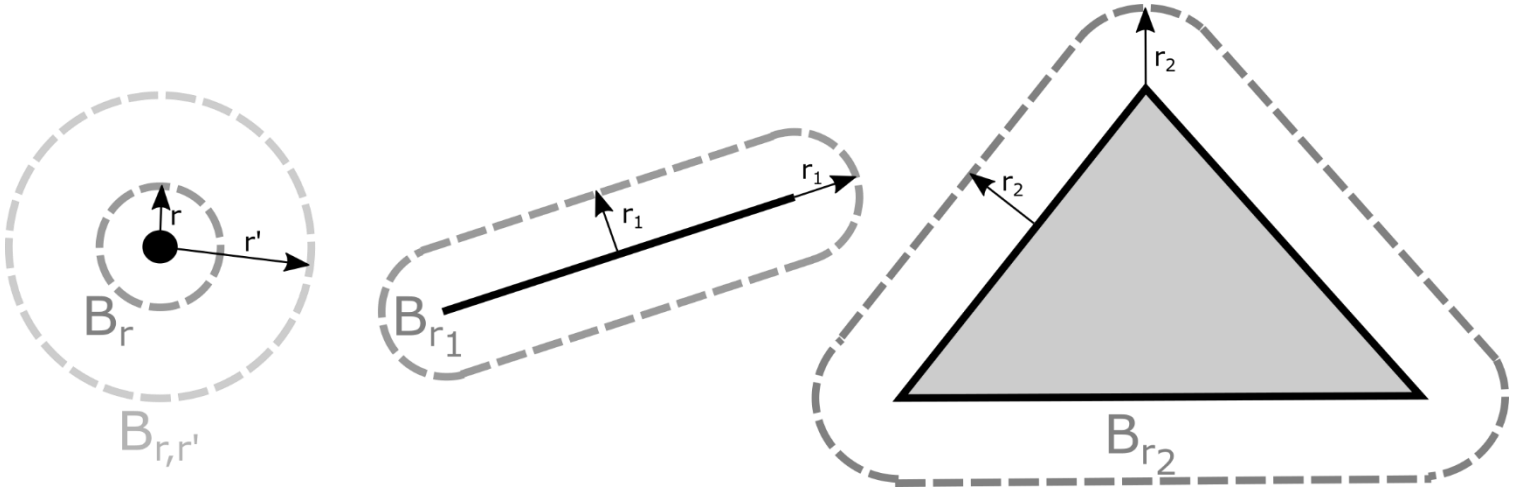
Problem

Gegeben sind mehrere Geoobjekte. Wie lassen sich dieses anhand ihres relativen Abstands klassifizieren?

- Geoobjekt O wird virtuell um einen Betrag r flächig erweitert. Dadurch entsteht eine s.g. *Pufferzone (*Zone relativer Nähe*, proximity zone, buffer zone)* $B_r(O)$. Es gilt:

$$x \in B_r(O) \rightarrow d(x, O) \leq r$$

- Pufferkoridor $B_{r,r'}(O)$: $x \in B_{r,r'}(O) \rightarrow r \leq d(x, O) \leq r'$



Buffering

Problem

Gegeben sind mehrere Geoobjekte. Wie lassen sich dieses anhand ihres relativen Abstands klassifizieren?

- Räumliche Klassifikation nur basierend auf senkrechtem Abstand $d(x, O)$
- Pufferzonen liegen nicht als eigene Objekte vor, werden aber ggf. visualisiert
- Erleichtert visuelle Interpretation durch sichtbare Abstandsgruppen

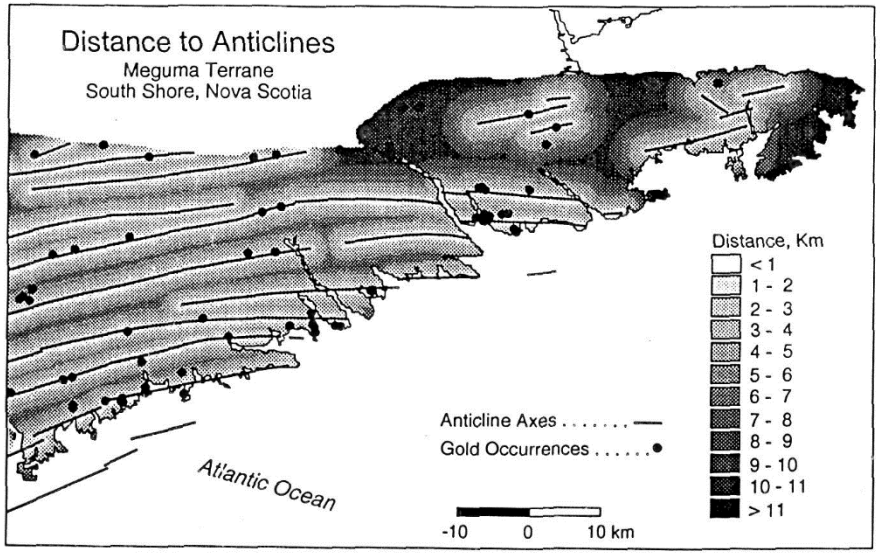


FIG. 6-8. A. Map to illustrate the dilation or buffering of linear features. Anticlinal fold axes in Meguma terrane, Nova Scotia have been successively dilated with corridors (250 m intervals) to produce a map showing proximity to the nearest fold axis. A classification has been applied so that the intervals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

Fläche-zu-Punkt Transformationen

Problem

Gegeben sind ein oder mehrere Flächenobjekte mit Attributen. Diese sollten jetzt an gegebenen Punktpositionen ausgewertet werden.

- Geometrisches Problem: **Punkt-in-Polygon-Test**
  Befindet sich ein Punkt innerhalb eines Polygons?

- Verschiedene Verfahren basierend auf verschiedenen Arten von Polygonen

  - Achs-paralleles Rechteck
  - Dreieck
  - Beliebiges Polygon

Fläche-zu-Punkt Transformationen

Bounding-Box-Test

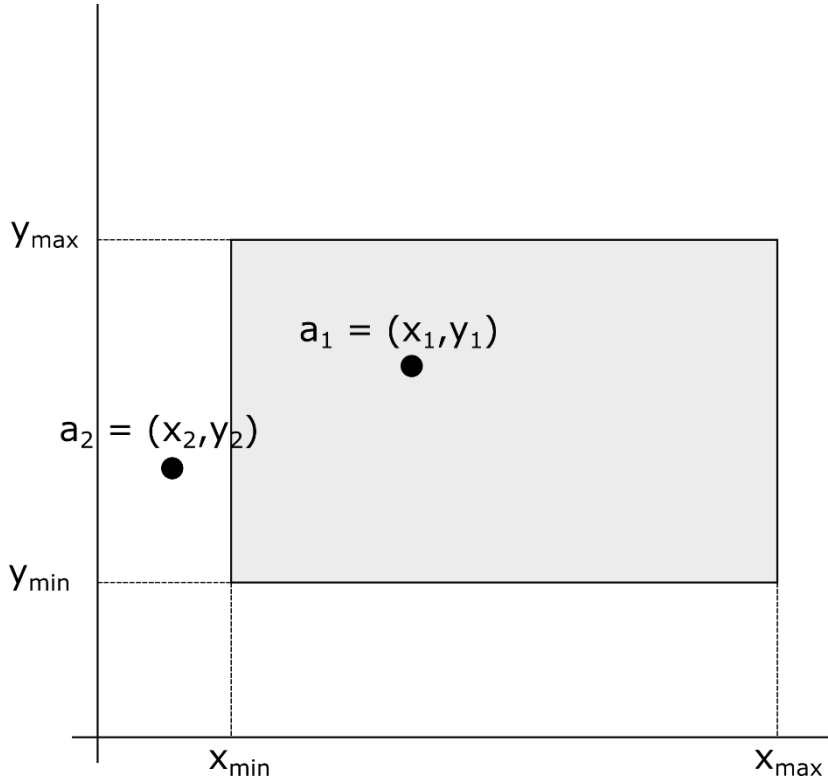
- Gegeben:
  Rechteck $R = (x_{min}, x_{max}, y_{min}, y_{max})$
  Abfragepunkt (x, y)

- Bedingung:

$$x \in \mathrm{R} \rightarrow \begin{cases} x_{min} \leq x \leq x_{max} \\ y_{min} \leq y \leq y_{max} \end{cases}$$

- Vorteil:
  - sehr effizient
  - leicht erweiterbar
    in höhere Dimensionen

- Nachteil: begrenzt anwendbar



- Anwendung:
  Identifizierung, ob ein Punkt in der AABB* eines Objektes liegt. Wenn ja, lohnen sich aufwändigere Verfahren für genauere Abfragen.

*axis-aligned bounding box

Fläche-zu-Punkt Transformationen

Punkt-in-Dreieck-Test

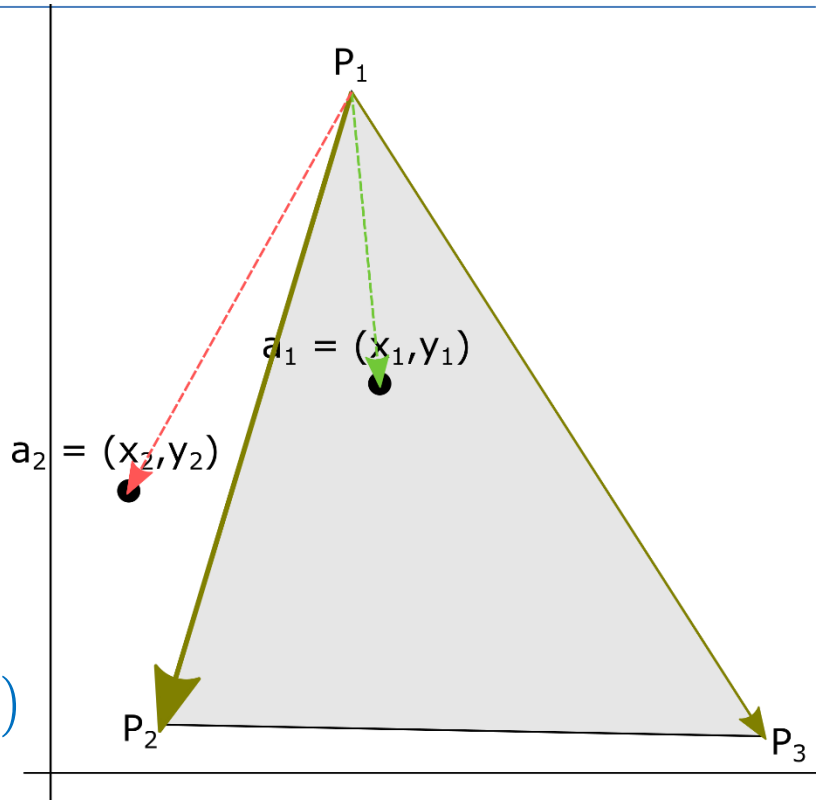
- Gegeben:
  Dreieck $\Delta(\vec{\boldsymbol{p}}_i, \vec{\boldsymbol{p}}_j, \vec{\boldsymbol{p}}_k)$
  Abfragepunkt $\vec{\boldsymbol{x}} = (x, y)$

- Ansatz:
  - jede Dreieckskante teilt den Raum in 2 Halbräume
  - liegt $\vec{\boldsymbol{x}}$ im gleichen Halbraum wie der der Kante $(\vec{\boldsymbol{p}}_i, \vec{\boldsymbol{p}}_j)$ gegenüberliegende Punkt $\vec{\boldsymbol{p}}_k$ liegt $\vec{\boldsymbol{x}}$ potentiell im Dreieck Δ

  ➢ $w_{ij}(\vec{\boldsymbol{x}}) > 0$ mit

$$w_{ij}(\vec{\boldsymbol{x}}) = \left((\vec{\boldsymbol{p}}_j - \vec{\boldsymbol{p}}_i) \times (\vec{\boldsymbol{p}}_k - \vec{\boldsymbol{p}}_i)\right) \cdot \left((\vec{\boldsymbol{p}}_j - \vec{\boldsymbol{p}}_i) \times (\vec{\boldsymbol{x}} - \vec{\boldsymbol{p}}_i)\right)$$

  ➢ $\vec{\boldsymbol{x}} \in \Delta \rightarrow w_{ij}(\vec{\boldsymbol{x}}) > 0,\ w_{jk}(\vec{\boldsymbol{x}}) > 0,\ w_{ki}(\vec{\boldsymbol{x}}) > 0$



- Test kann abgebrochen werden, wenn **ein** $w < 0$ gefunden wird

- Vorteil: Berechnung leicht optimierbar
- Nachteil: nur für Dreiecke geeignet

Fläche-zu-Punkt Transformationen

Punkt-in-Polygon-Test / Strahlmethode nach Jordan

- Gegeben:
  Polygon $\boldsymbol{P}(\vec{\boldsymbol{p}}_1, \vec{\boldsymbol{p}}_2, \ldots, \vec{\boldsymbol{p}}_n)$ mit Sequenz aus n Randpunkten
  ⇒ Kante $k_i = \vec{\boldsymbol{p}}_{i+1} - \vec{\boldsymbol{p}}_i, 1 \leq i \leq n-1$; $k_n = \vec{\boldsymbol{p}}_1 - \vec{\boldsymbol{p}}_n$

  Abfragepunkt $\vec{\boldsymbol{x}} = (x, y)$

- Ansatz:
  1. Definiere einen Strahl mit beliebiger Richtung $\vec{r}$, ausgehend von $\vec{\boldsymbol{x}}$
  2. Teste für jede Kante, ob der Strahl die Kante schneidet
     $$\forall k_i: \vec{\boldsymbol{x}} + \lambda_r \vec{r} = \vec{\boldsymbol{p}}_i + \lambda_i (\vec{\boldsymbol{p}}_{i+1} - \vec{\boldsymbol{p}}_i)$$
     - Schnitt wenn $\lambda_i \geq 0$ und $\lambda_i < 1$
  3. Zähle alle Schnitte zwischen Strahl und Kanten: n_h

- n_h ist **ungerade**: $\vec{\boldsymbol{x}} \in \boldsymbol{P}$
- n_h ist **0 oder gerade**: $\vec{\boldsymbol{x}} \notin \boldsymbol{P}$

Fläche-zu-Punkt Transformationen

Punkt-in-Polygon-Test / Strahlmethode nach Jordan

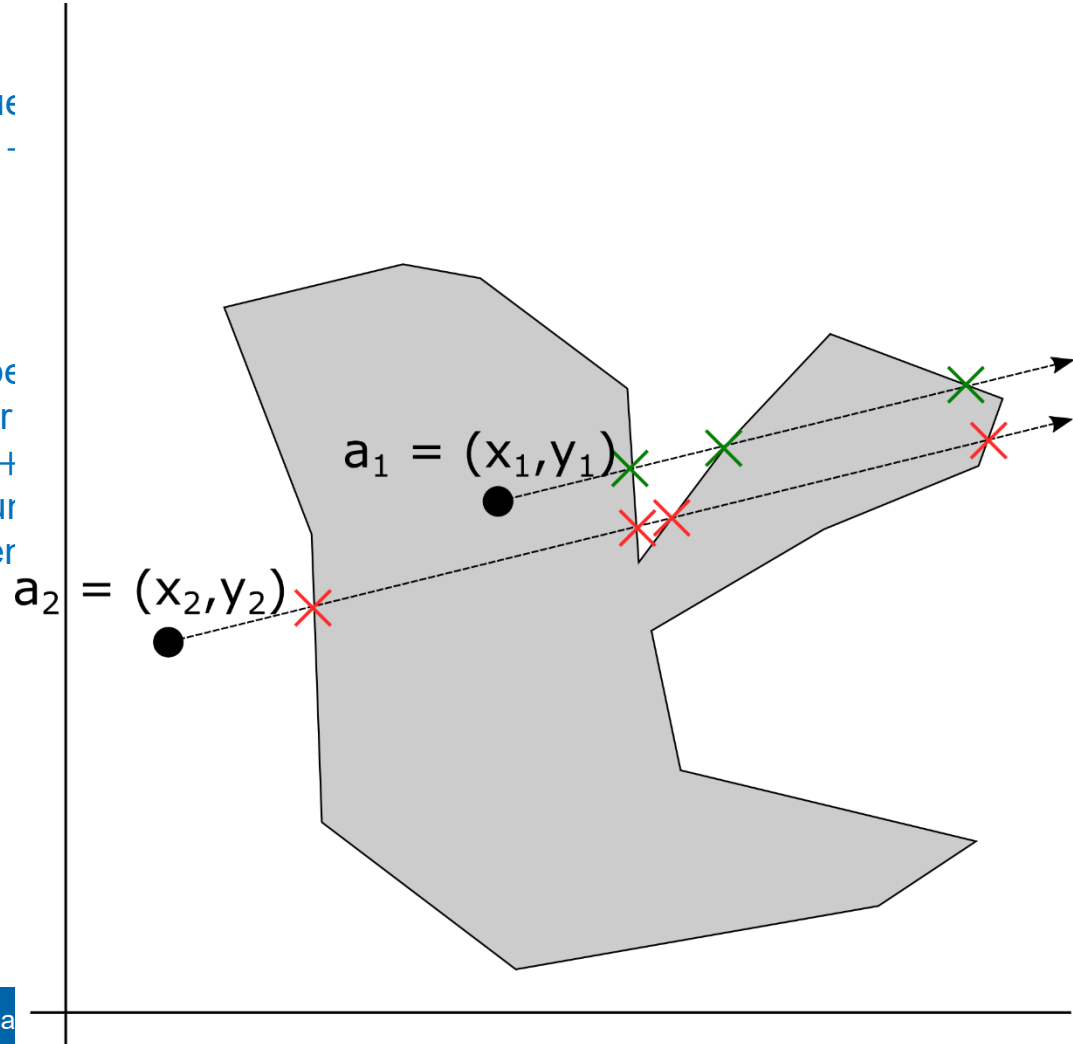
- Gegeben:
  Polygon $\boldsymbol{P}(\vec{\boldsymbol{p}}_1, \vec{\boldsymbol{p}}_2, \ldots, \vec{\boldsymbol{p}}_n)$ mit Seque
  $\Rightarrow$ Kante $k_i = \vec{\boldsymbol{p}}_{i+1} - \vec{\boldsymbol{p}}_i, 1 \leq i \leq n$ -

  Abfragepunkt $\vec{\boldsymbol{x}} = (x, y)$

- Ansatz:
  1. Definiere einen Strahl mit be
  2. Teste für jede Kante, ob der
     $\forall k_i$: $\vec{\boldsymbol{x}} + \lambda_r \vec{r} = \vec{\boldsymbol{p}}_i +$
     - Schnitt wenn $\lambda_i \geq 0$ ur
  3. Zähle alle Schnitte zwischer

- n_h ist **ungerade**: $\vec{\boldsymbol{x}} \in \boldsymbol{P}$
- n_h ist **0 oder gerade**: $\vec{\boldsymbol{x}} \notin \boldsymbol{P}$

- Vorteil: funktioniert für **alle** Polygone
- Nachteil: aufwändig, es müssen immer **alle** Kanten getestet werden

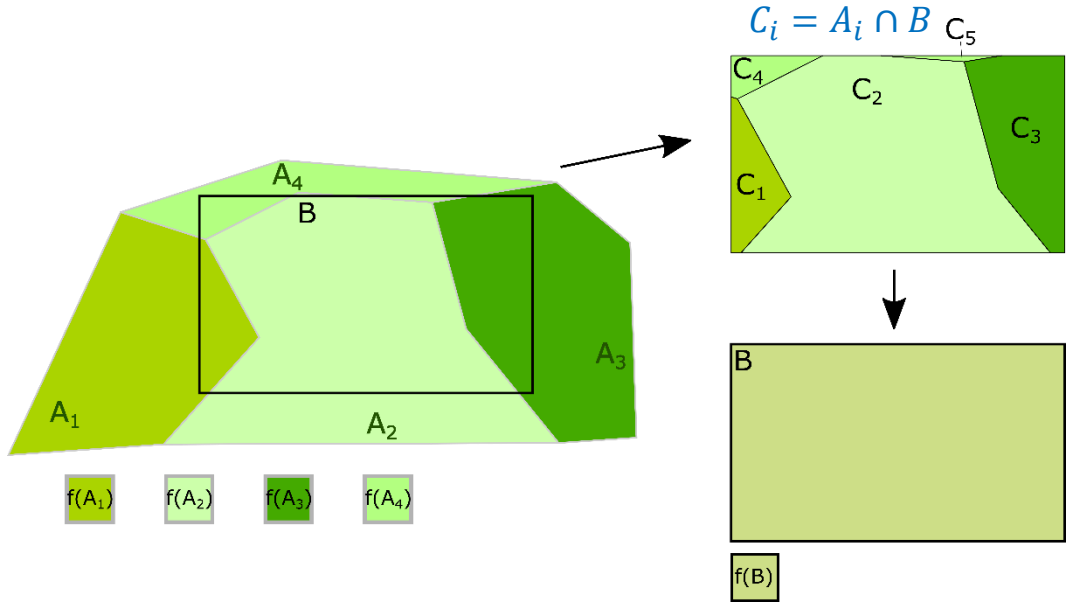


Fläche-zu-Fläche Transformation

Problem

Gegeben sind ein oder mehrere Flächenobjekte mit Attributen. Diese sollten jetzt für ein gegebenes anderes Flächenobjekt ausgewertet werden.

- Gegeben: Menge von Polygonen $(A_i, f(A_i))$ und ein Abfragepolygon B
- Gesucht: $f(B) = \sum_i \lambda_i f(A_i)$
  - $\lambda_i = \frac{\text{Fläche}(A_i \cap B)}{\text{Fläche}(B)}$



Resampling Transformation

upsampling

- Erhöhung der Auflösung
  - Hinzufügen von zusätzlichen unabhängigen Daten 
  - Interpolation …? → keine zusätzliche Information

downsampling

- Verringerung der Auflösung
  - Interpolation …? → Gefahr von Informationsverlust
  - **weeding**

Resampling Transformation

weeding

- Entfernen von Datenpunkten mit dem geringsten Beitrag zum Informationsgehalt
- basierend auf Geometrie und/oder Parameterverteilung

➢ Ziel: verbleibende Datenpunkte umfassen die relevante Information

➢ Beispiel: **Douglas-Peucker-Algorithmus zur Ausdünnung von Liniendaten**

Resampling Transformation

Douglas-Peucker-Algorithmus zur Ausdünnung von Liniendaten

1. Verbinde Anfangs- und Endpunkt mit einer geraden Linie

2. Füge den Datenpunkt ein, der den größten senkrechten Abstand zur Verbindungslinie
   - ➢ Zwei neue Segmente

3. Wiederhole Punkt 2 für jedes neue Segment bis kein Datenpunkt ist weiter entfernt als gegebenes Minimum

- ➢ Datenpunkte, welche nicht eingefügt werden, tragen zu wenig Information zur Liniengeometrie bei

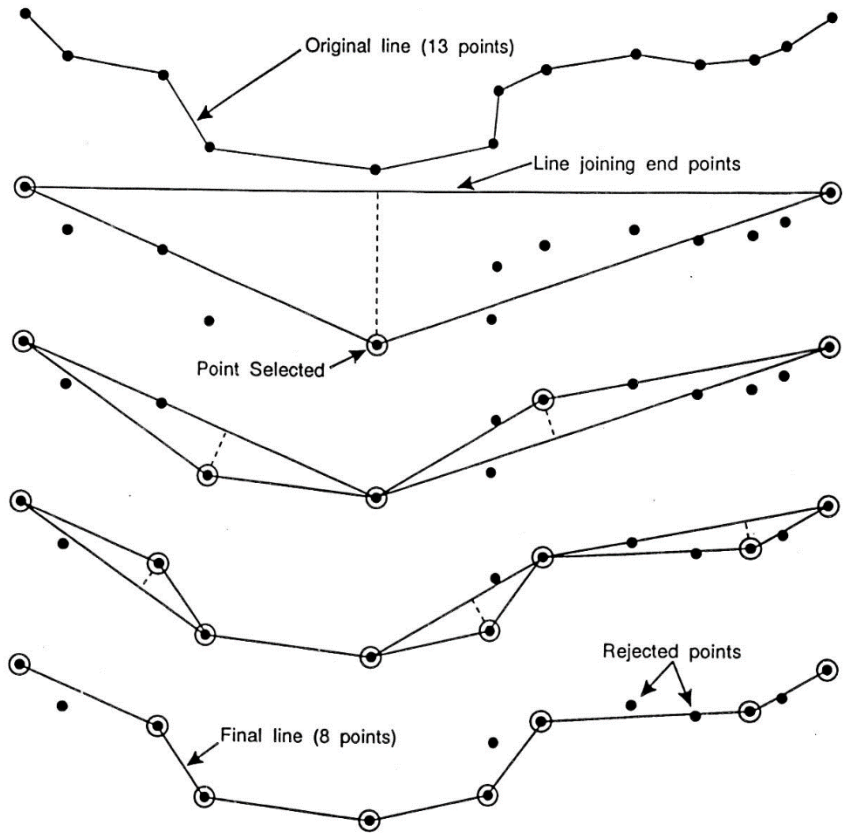


FIG. 6-10. Douglas-Peucker algorithm for line generalization. The original line contains 13 points, and the final line contains 8 points. The rejected points are said to be "weeded". A number of alternative algorithms for line weeding have been proposed.

Übersicht

1. Definitionen, Funktionen, Anwendungen
2. Koordinatensysteme und -transformationen
3. Räumliche Datenmodellierung
4. Vermaschungen
5. Räumliche Interpolation
6. Transformationen, Filtermethoden
7. **Geodatenmanagement**

Geodatenmanagement und GIS Projekte

Grundsätzliche verschiedene Arten von GIS Projekten:

1. „*Single Task*“-Projekt:
   - Bearbeitet eine einzelne, spezifische Problemstellung, z.B.
     - Kombination mehrere Eingabeobjekte zu einem Ausgabeobjekt
     - Digitalisierung einer Feature Class basierend auf einem Rasterobjekt
   - Eher klein, wenige beteiligte Geoobjekte
   - Nur ein Anwender

2. Komplexes Einzelprojekt:
   - Bearbeitet eine komplexe Problemstellung, welche die Bearbeitung mehrerer spezifischer „Unterprobleme“ voraussetzt
   - Mittlerer Datenumfang, zum Teil zusätzliche „Zwischenobjekte“
   - Zumeist nur ein Anwender, ggf. unabhängige Nutzer

3. „*Multi-Task*“- oder „*Multi-Purpose*“-Projekt:
   - Bearbeitung mehrerer komplexer Problemstellung basierend auf einer gemeinsamen Datenbasis, zum Teil aufgeteilt in mehrere einzelne GIS-Projekte
   - Umfangreiche Datenbasis
   - Zumeist mehrere Anwender und unabhängige Nutzer

Geodatenmanagement und GIS Projekte

Welche Datenquellen sollte man verwenden:

1. „*Single Task*“-Projekt:
   - Einzeldateien (z.B. *shape-Files, GeoTIFF*), Geodatenbank

2. Komplexes Einzelprojekt:
   - Geodatenbank
   - Eingangsdaten sollten ebenfalls per Datenbank vorgehalten werden
   - Ggf. Export einzelner Datensätze als Einzeldateien

3. „*Multi-Task*“- oder „*Multi-Purpose*“-Projekt:
   - Mehrere Geodatenbanken (Datenbasis thematisch aufgeteilt), ggf. mit *Web Map Server* (WMS)-Anschluss

Wie sollten Geoobjekte benannt werden?

Was macht einen guten Objektnamen aus?

- Gemäß existierend Konventionen
- Möglichst **deskriptiv** („beschreibend“)
- Möglichst **eindeutig**

➢ **Für Dritte interpretierbar!**
➢ **Übersichtlichkeit der verwendeten Datenquelle!**

Sollte der Objekttyp teil des Objektnamens sein?

- **Nein**, für Objekte welche nur innerhalb klassischer GIS-Projekte verwendet wird

  ➢ GIS-Software erkennt den Typ automatisch und visualisiert diesen

Wie sollten Geoobje

Was macht einen guten Objektnamen au

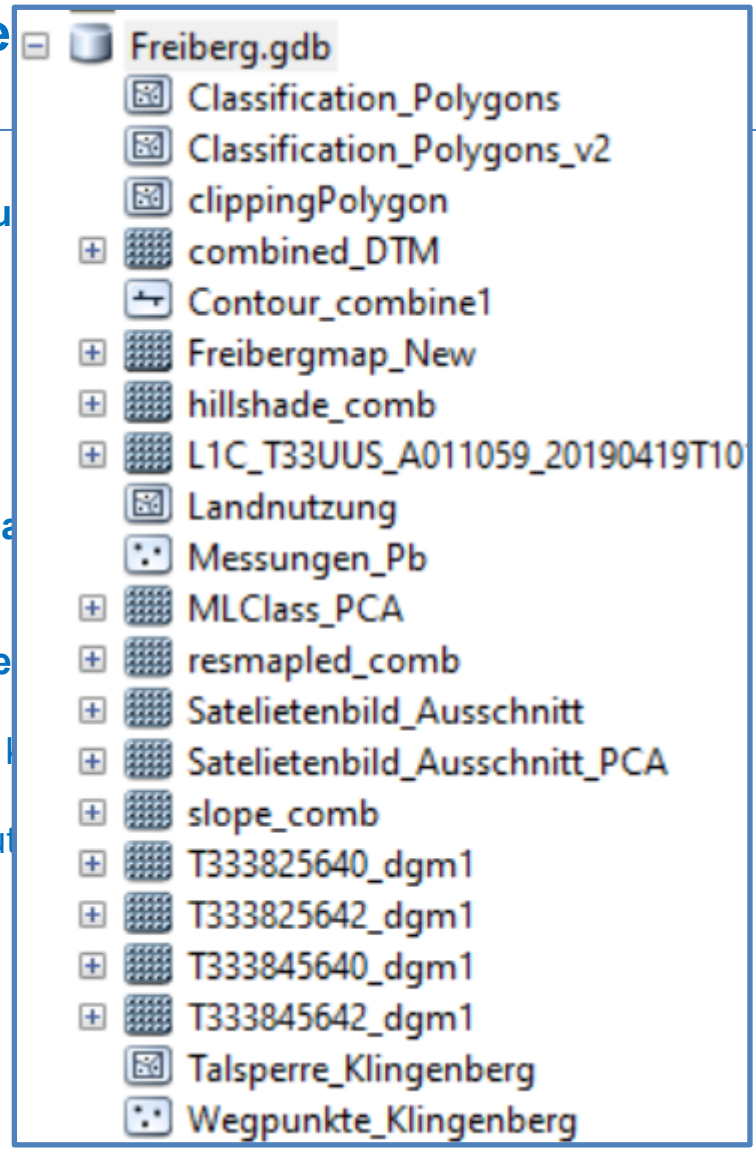
- Gemäß existierend Konventionen
- Möglichst **deskriptiv** („beschreibend“)
- Möglichst **eindeutig**

➢ **Für Dritte interpretierbar!**
➢ **Übersichtlichkeit der verwendeten Da**

Sollte der Objekttyp teil des Objektname

- **Nein**, für Objekte welche nur innerhalb

  ➢ GIS-Software erkennt den Typ au



Wie sollten Geoobjekte benannt werden?

Was macht einen guten Objektnamen aus?

- Gemäß existierend Konventionen
- Möglichst **deskriptiv** („beschreibend“)
- Möglichst **eindeutig**

➢ **Für Dritte interpretierbar!**
➢ **Übersichtlichkeit der verwendeten Datenquelle!**

Sollte der Objekttyp teil des Objektnamens sein?

- **Nein**, für Objekte, welche nur innerhalb klassischer GIS-Projekte verwendet wird.
  - ➢ GIS-Software erkennt den Typ automatisch und visualisiert diesen.
- Für Objekte, welche auch außerhalb von dezidierter GIS-Software verwendet werden, kann die Angabe des Typs im Name ggf. sinnvoll sein.

Wie sollten Geoobjekte benannt werden?

Was macht einen guten Objektnamen aus?

- Gemäß existierend Konventionen
- Möglichst **deskriptiv** („beschreibend“)
- Möglichst **eindeutig**

➢ **Für Dritte interpretierbar!**
➢ **Übersichtlichkeit der verwendeten Datenquelle!**

Sollte der Objekttyp teil des Objektnamens sein?

- **Nein**, für Objekte, welche nur innerhalb klassischer GIS-Projekte verwendet wird.
  - ➢ GIS-Software erkennt den Typ automatisch und visualisiert diesen.
- Für Objekte, welche auch außerhalb von dezidierter GIS-Software verwendet werden, kann die Angabe des Typs im Name ggf. sinnvoll sein.

Objektbenennung ist nicht statisch, d.h. im Laufe der Bearbeitung ist es möglich, ein Objekt umzubenennen!

Metadaten für Geoobjekte

Was sind Metadaten?

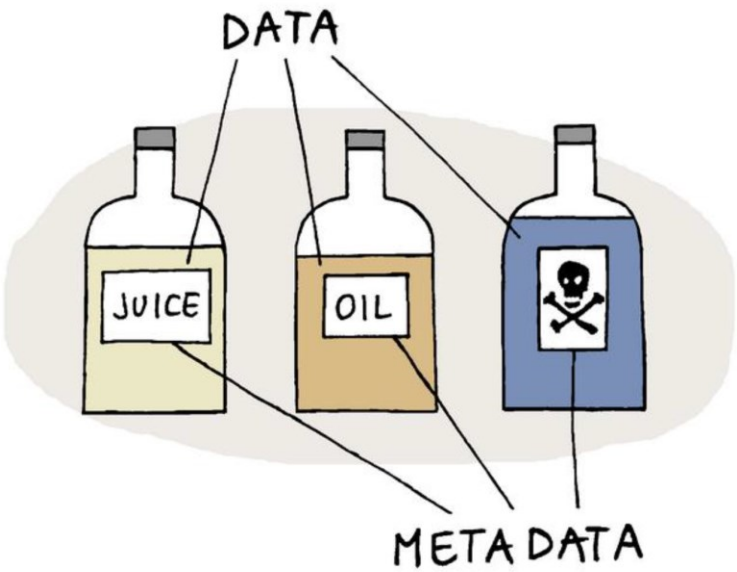


Image: https://dataedo.com/cartoon/data-vs-metadata-4

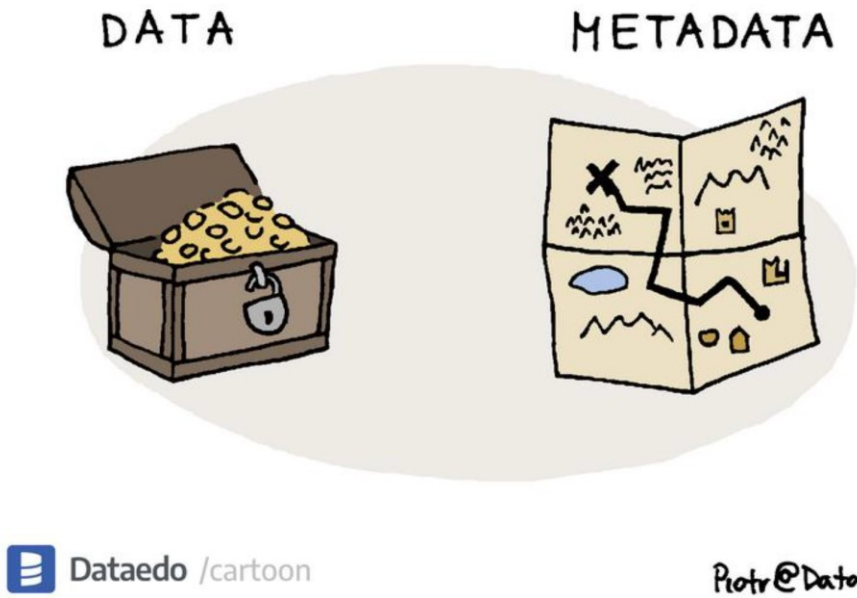


Image: https://dataedo.com/cartoon/data-vs-metadata-2

Metadaten für Geoobjekte

Was sind Metadaten …

… „Daten über Daten“ / Datendokumentation / …

- **Deskriptiv:** Informationen über den Inhalt
  - *Objektname, Ersteller, Spatial Reference, Erstellungszeit, Zweck, Beschreibung …*
- **Administrativ:** Informationen zu Verwaltungszwecken
  - *Datenquelle / Datenformat, Version, Copyright / Lizenz …*
- **Strukturell:** Informationen zu Beziehungen zwischen Objekten
  - *Zugrundeliegende Objekte, untergeordnete Objekte …*

ArcGIS erlaubt es rudimentär, Metainformationen für einzelne Objekte hinzuzufügen. Dies muss aber durch den Anwender selbst geschehen und die Informationen lassen sich nur schwer von „außerhalb“ auslesen.

➢ **Im Allg. muss der Anwender die Metainformationen von Geoobjekten selbst verwalten. Dies kann aufwändig sein, ist aber notwendig …**

FAIR-Prinzipien

- **F**indable (Auffindbarkeit)
  - Wo sind die Daten abgelegt?
- **A**ccessible (Zugreifbarkeit / Verfügbarkeit)
  - Wie kann ich auf die Daten zugreifen?
- **I**nteroperability (generelle Verwendbarkeit)
  - Wie können die Daten verwendet werden?
- **R**eusability (Wiederverwendbarkeit)
  - Wie können andere die Daten verwenden?

Im Allgemeinen bzgl. Öffentlich publizierten Daten, aber auch für lokale Daten sinnvoll.

➢ Umfassende Dokumentation und ausführliche Metadaten sind **IMMER** sinnvoll und notwendig!

Wilkinson, M., Dumontier, M., Aalbersberg, I. *et al.*(2016). The FAIR Guiding Principles for scientific data management and stewardship. *SciData***3,** 160018. https://doi.org/10.1038/sdata.2016.18

FAIR-Prinzipien: Findable

Öffentlich:

- Global eindeutiger und persistenter Identifizierer (z.B. DOI)
- Ausführliche / komplette Metadaten
- Metadaten enthalten Verweis zum Identifizierer der Daten
- Daten liegen in einer öffentlich durchsuchbaren Quelle (Datenbank) vor oder sind dort verlinkt

Lokal:

- Daten sind zumindest eindeutig benannt
- Ausführliche Metadaten / Dokumentation, wo sich alle Daten befinden

Wilkinson, M., Dumontier, M., Aalbersberg, I. *et al.*(2016). The FAIR Guiding Principles for scientific data management and stewardship. *SciData***3,** 160018. https://doi.org/10.1038/sdata.2016.18

FAIR-Prinzipien: Accessible

Öffentlich:

- (Meta)Daten können über den Identifizierer mittels eines standardisierten Übertragungsprotokolls abgerufen werden
- Das verwendete Protokoll ist öffentlich, frei verfügbar und universell implementierbar
- Wenn die Daten nicht verfügbar sind, sind zumindest die Metadaten verfügbar

Lokal:

- Daten und Metadaten sollten getrennt von einander archiviert werden

Wilkinson, M., Dumontier, M., Aalbersberg, I. *et al.*(2016). The FAIR Guiding Principles for scientific data management and stewardship. *SciData***3,** 160018. https://doi.org/10.1038/sdata.2016.18

FAIR-Prinzipien: Interoperatibilty

Öffentlich:

- (Meta)Daten liegen in einer formalen, zugänglichen, teilbaren und weit verbreitenten „Sprache“ / Repräsentation vor (*XML*, *JSON*,…)
- (Meta)Daten verwenden Vokabular / Bezeichungen, welche den FAIR-Prinzipien entsprechen
- (Meta)Daten verweisen nur über offizielle eindeutige Identifierer (fully qualified reference) auf andere (Meta)Daten

Lokal:

- Alle Metadaten werden gemäß einer eindeutigen Konvention erstellt
- Alle Metadaten liegen in einfach-lesbarer / leicht-verarbeitbarer Form vor (ASCII-Dateien, XML)

Wilkinson, M., Dumontier, M., Aalbersberg, I. *et al.*(2016). The FAIR Guiding Principles for scientific data management and stewardship. *SciData***3,** 160018. https://doi.org/10.1038/sdata.2016.18

FAIR-Prinzipien: Reusable

Öffentlich:

- (Meta)Daten beinhalten **ALLE relevanten Attribute**
- (Meta)Daten werden mittels klaren und eindeutigen Lizensierung veröffentlicht
- (Meta)Daten beinhalten detaillierte Herkunftsinformationen / provenance data.
- (Meta)Daten den Standards des betreffenden Fachgebiets

Lokal:

- Alle (Meta)Daten werden, wenn möglich, in nicht-proprietären Datenformaten als Quasistandards / Standardformate vorgehalten / archiviert (shape-files / Pdf / ASCII-Dateien)
- Alle Daten sind ausführlich dokumentiert

Wilkinson, M., Dumontier, M., Aalbersberg, I. *et al.*(2016). The FAIR Guiding Principles for scientific data management and stewardship. *SciData***3,** 160018. https://doi.org/10.1038/sdata.2016.18

… Reproduzierbarkeit?

Ist es möglich, bei nochmaliger Bearbeitung einer Fragestellung mit der gleichen Datenbasis zu den gleichen Ergebnissen zu kommen?

Sollten Geodaten / Geoobjekte aus GIS-Projekten reproduzierbar sein?

- **Ja,** zumindest im Falle von Risikoabschätzung und Gefahrenabwehr!

Sind solche Geodaten / Geoobjekte überhaupt reproduzierbar?

- Herkunftsdaten sind auf notwendig …
  Wer hat **mit welchen Daten** die Fragestellung **WIE** bearbeitet?
- Wie kann man interaktive Arbeitsschritte reproduzierbar machen?
  - *Logging* / *History* – Daten
- Erstellung eines Code-Skripts, welche alle Arbeitsschritte umfasst.
  - Für ArcGIS ⇒ Python-Schnittstelle ArcPy ist vorhanden.