TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG

Die Ressourcenuniversität. Seit 1765.

Grundlagen GIS

Summer Term 2022

Peter Menzel
Arbeitsgruppe Geomathematik und Geoinformatik

OPAL:
https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/197584157 3?6

Übersicht

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

Logistics

1. Wöchentliche Vorlesung – 15 Termine bis zum 13. Juli
2. Wöchentliche Übungen (BGIP, MA) oder 2-wöchentliche Übungen (BGÖK, BGM)
    1. Start in der Woche vom 11. bis 15. April
    2. PC-Pool MEI-1203a

       Zugang mit Studentenausweis
    3. ArcGIS kann auf privater

       Hardware verwendet werden

       (siehe OPAL / Übungen)

Zugang PC-Pool MEI-1203a

Freischaltung Dienst- bzw. Studentenausweis

E-Mail

An	maja.merz@geo.tu-freiberg.de
CC	tino.beyer@geophysik.tu-freiberg.de
Betreff	Freischaltung PC-Pool MEI-1203a + Haustür

- Name, Vorname
- Personal-/Matrikelnummer
- Bereich/Studiengang
- Dienstvorgesetzter/Betreuer
- Befristung

HINWEIS:

Die Freischaltung erfolgt zentral! Bei Funktionsstörungen wenden Sie sich bitte an die **Leitzentrale, Karl-Kegel-Bau, Agricolastraße 1, Zimmer: EG 42**

Logistics

- Abschluss des Moduls mittels 90 Min. Klausur (aktuell geplant in Präsenz)
- Zusatzpunkte für Klausur können über Übungsbelegt erreicht werden:
  - 90 Minuten am jeweils letzten Übungstermin einer Übungsgruppe
  - Umfasst: grundlegende Arbeiten in ArcGIS bis zum Erstellen einer druckbaren Karte

Logistics

- OPAL-Kurs:
  https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873
  - Wichtige aktuelle Informationen, Forum, …
  - Vorlesungsunterlagen + Online-Skript
  - Übungsunterlagen (gruppenabhängig)
  - Virtuelles Klassenzimmer

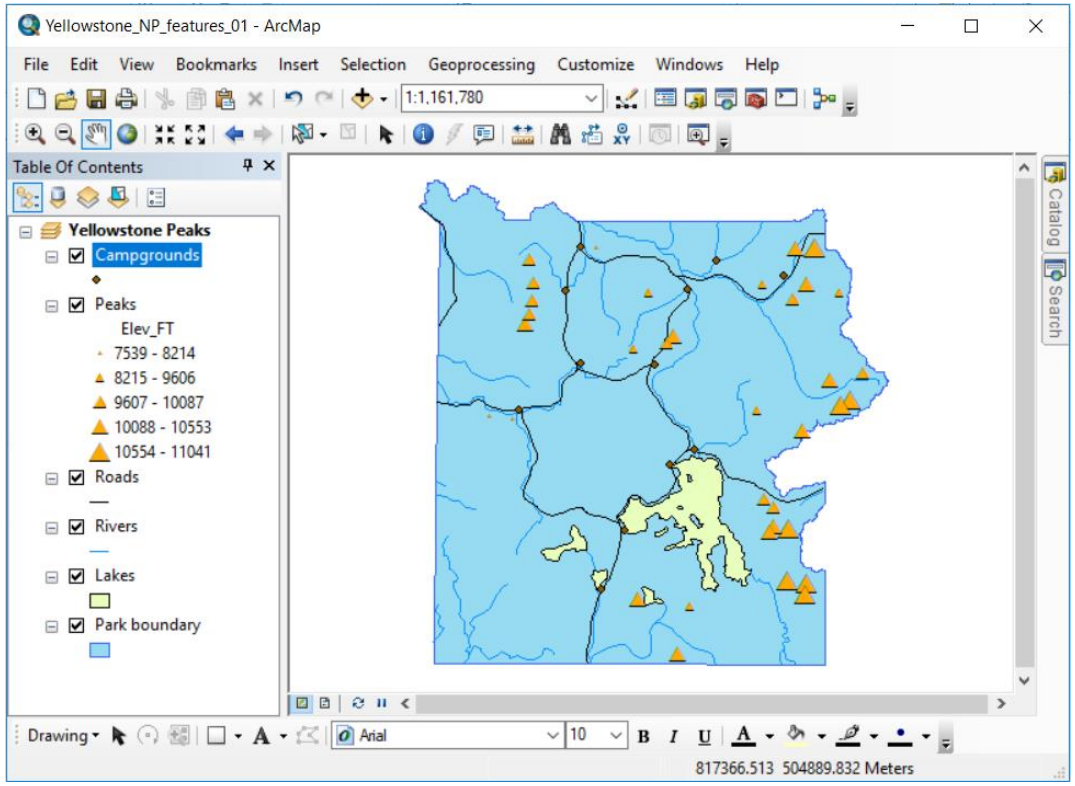
- Bitte tragen Sie sich in die OPAL-Lerngruppe für die Vorlesung ein:
  https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/164748769582796200 9

- Bitte tragen Sie sich in die OPAL-Lerngruppe für Ihre Übungsgruppe ein:
  https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/164757415654823000 3

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 Sach-verhalten)
- 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

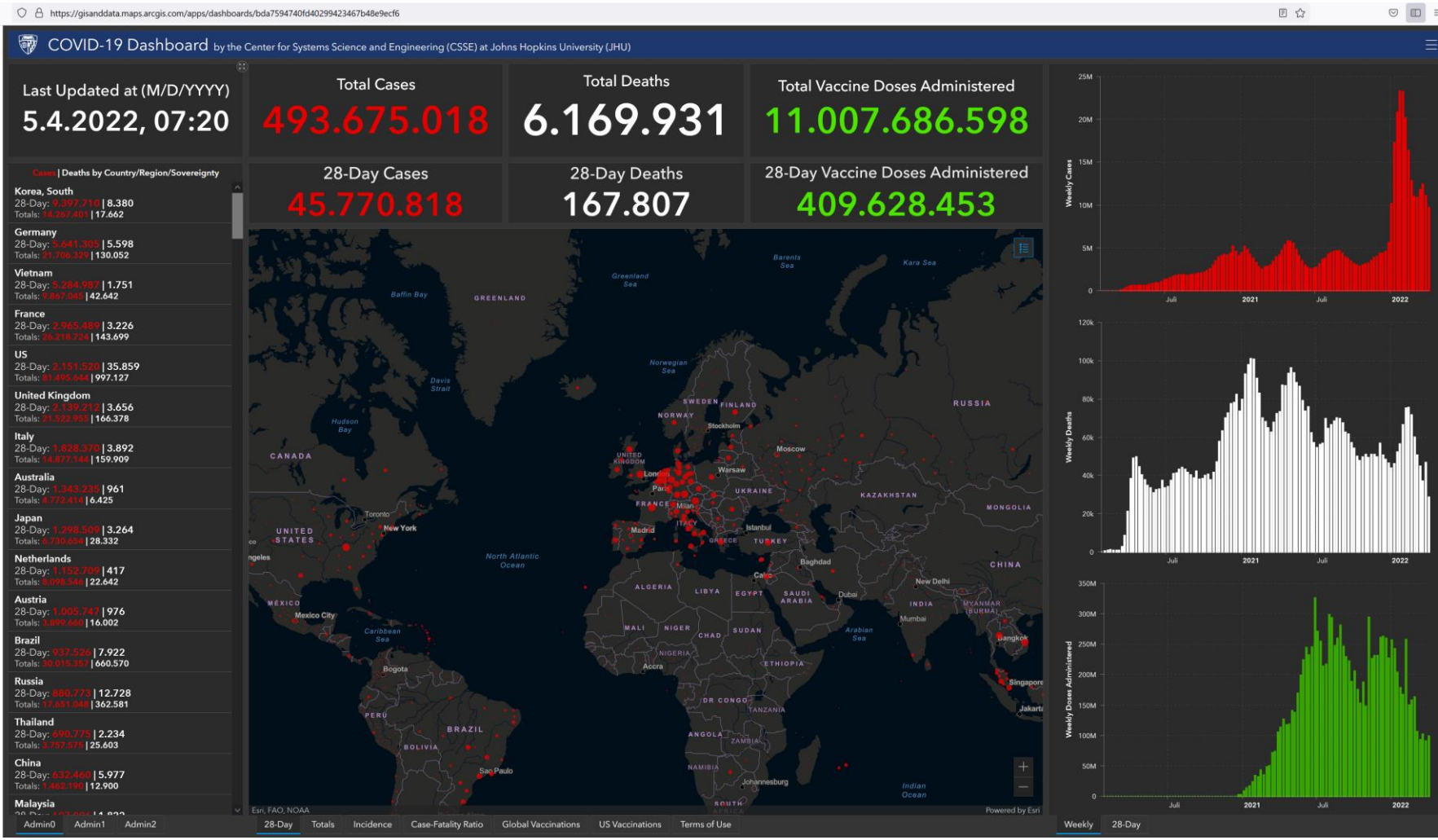
- Eingabe/Input (manuelle oder automatisierte Eingabe von Rohdaten)
- Management/Verwaltung (Aufbau von Datenbanken und Schnittstellen, sowie Bearbeitung von Geoobjekten)
- Analyse/Auswertung (Erstellung und Auswertung von Modellen zu gegebenen Sachverhalten)
- Präsentation

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: 6.710.656 | 28.332
Netherlands
28-Day: 1.152.709 | 417
Totals: 3.098.548 | 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: 4.757.575 | 25.603
China
28-Day: 632.460 | 5.977
Totals: 1.462.190 | 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

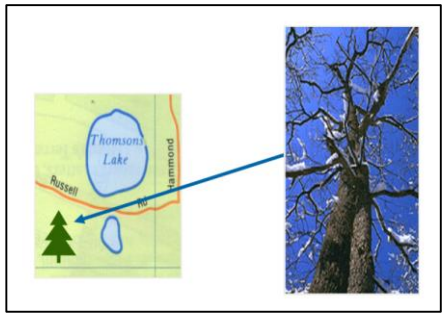
Daten- und Objektypen

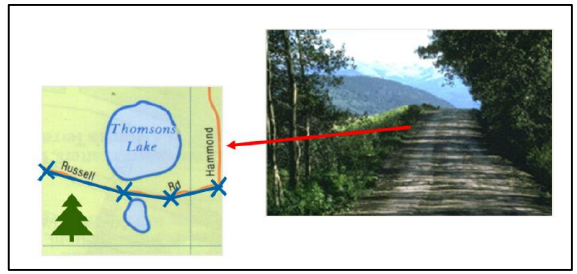
- **Geodaten vs. Sachdaten**:
  - Geodaten weisen stets einen Raumbezug auf und können in verschiedenen Datenmodellen (z.B. Vektor- oder Rasterdaten) vorliegen; es wird oft zwischen Geometriedaten und Topologiedaten unterschieden
  - Sachdaten liegen initial ohne Raumbezug vor (Namen, Nummern, Bezeichnungen,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

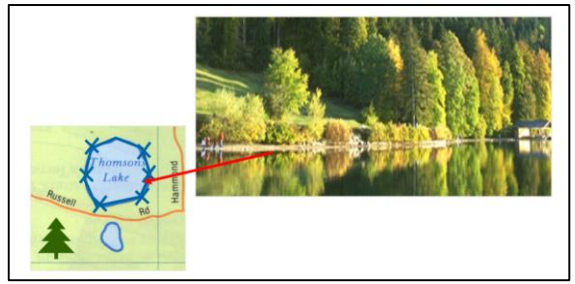
Daten- und Objektypen

- **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.)

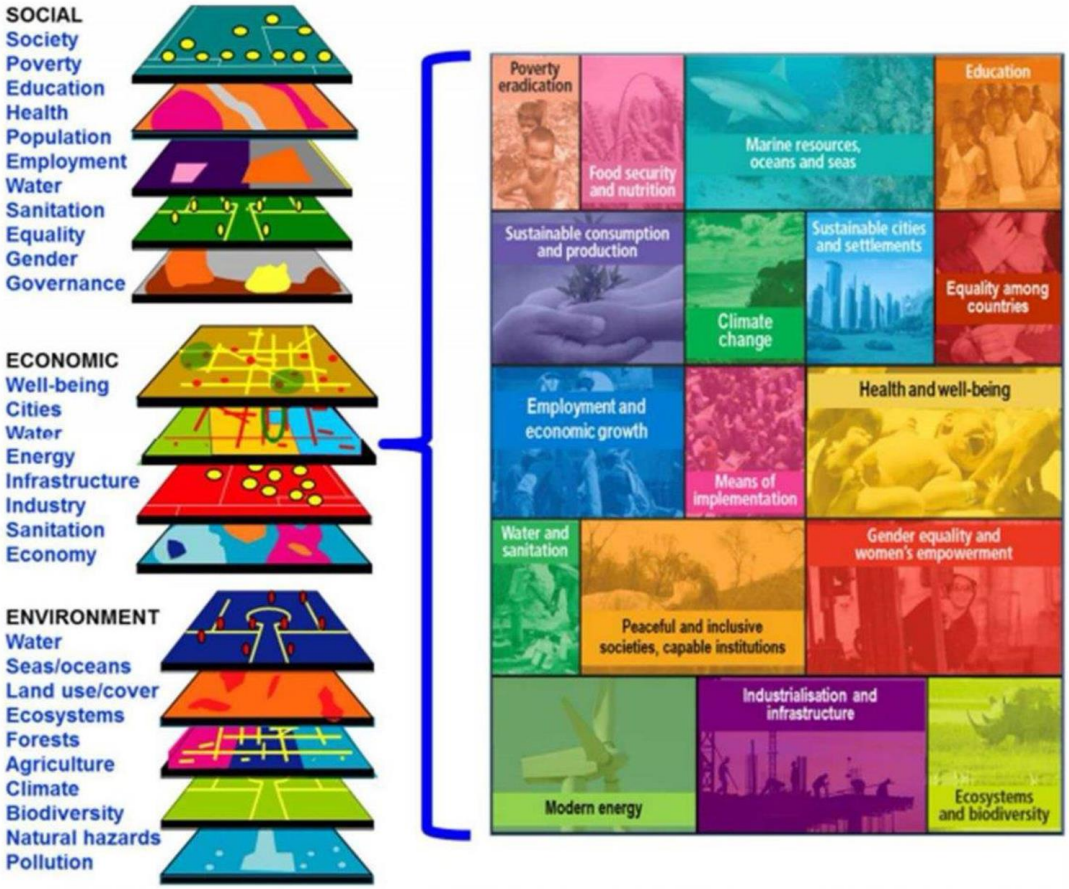
- **Geoobjekte**: Gruppierung von Daten in zusammenhängende, räumlich abgeschlossene Objekte mit ähnlichen Eigenschaften
  - Punkteobjekte (0-d)
  - Linienobjekte (1-d)
  - Flächenobjekte (2-d)
  - Volumenobjekte (3-d)
  - etc.





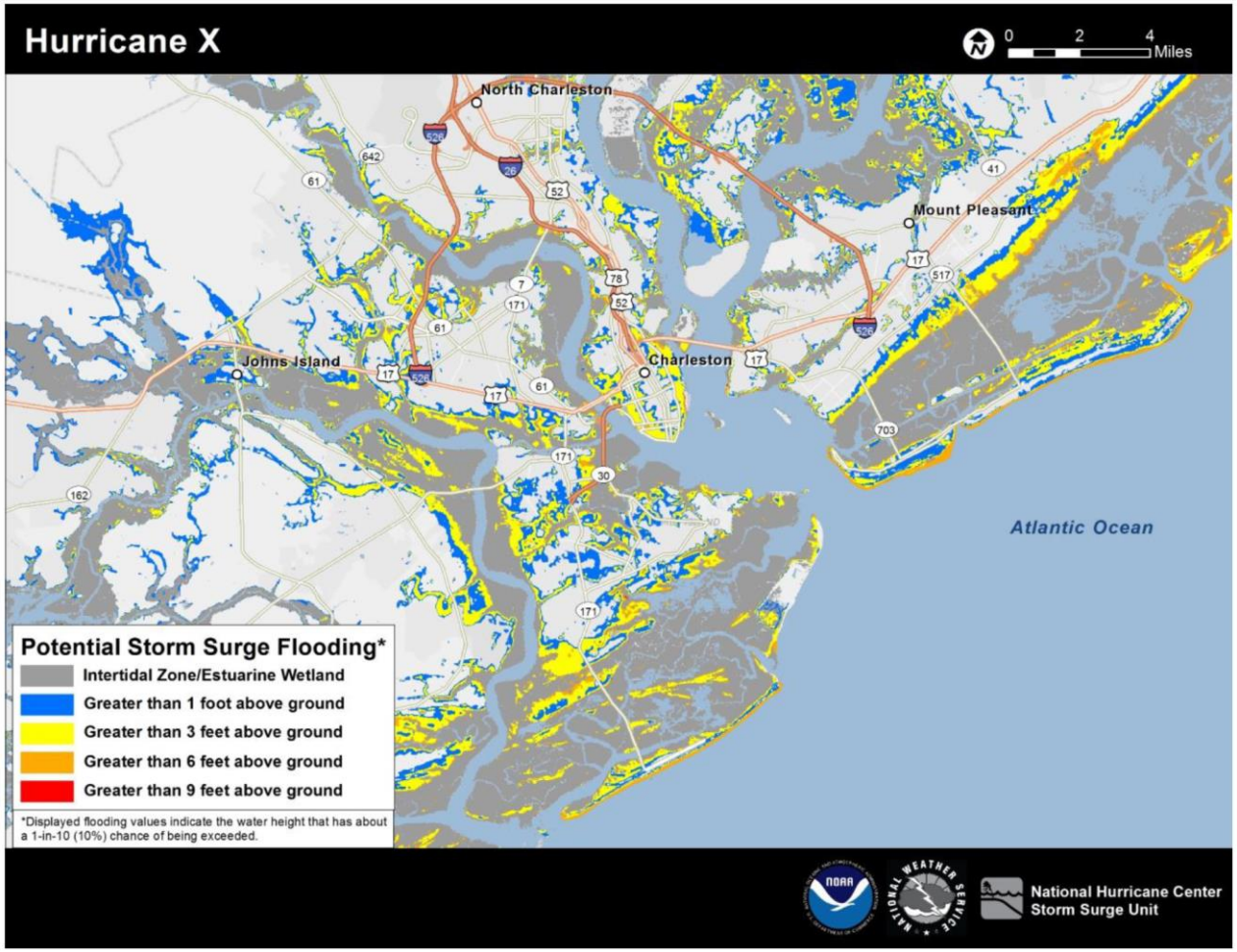


Daten- und Objektypen



Daten- und Objektypen

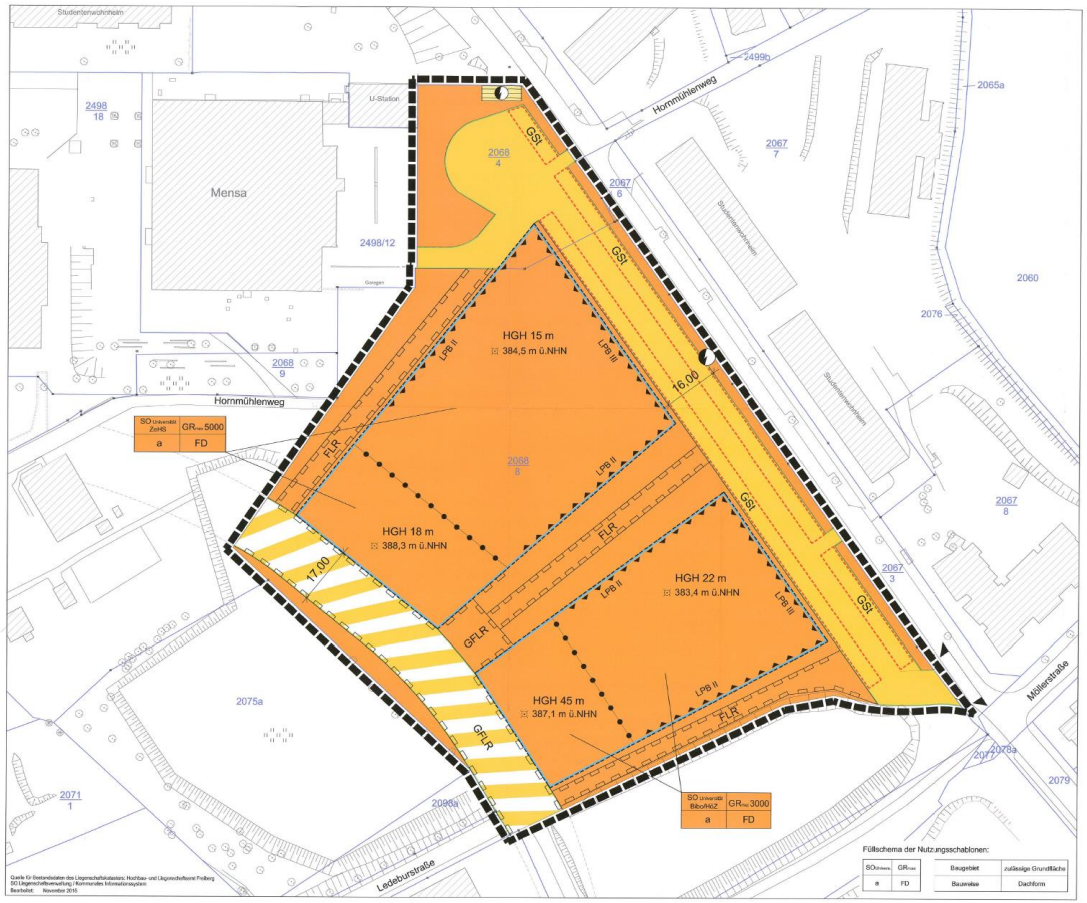
Primär- oder Sekundärdaten?



Daten- und Objektypen

Primär- oder Sekundärdaten?

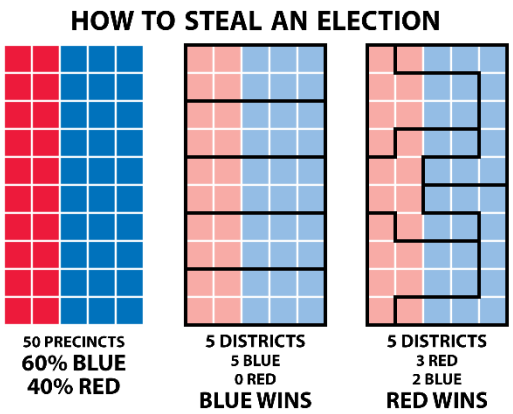
Vorhabenbezogener Bebauungsplan Nr. V 020 "Campuserweiterung am Wissenschaftskorridor"



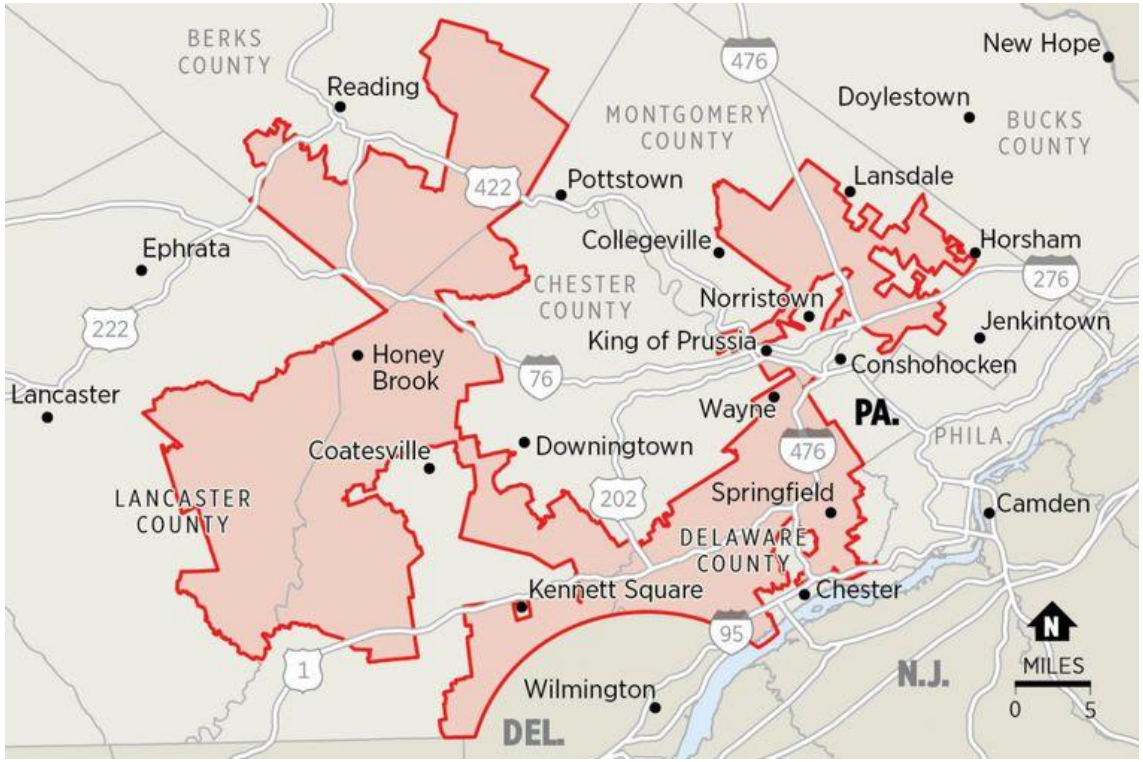


Daten- und Objektypen

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

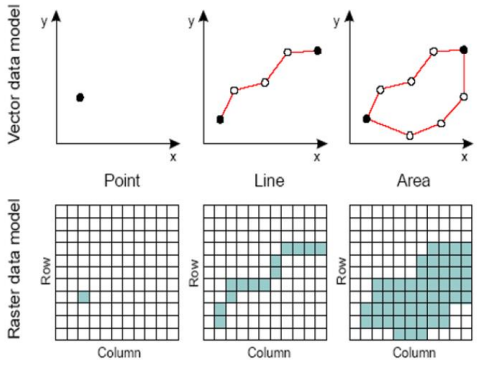
Funktionen eines GIS

Ein GIS Projekt unterteilt sich im Wesentlichen in drei Abschnitte:

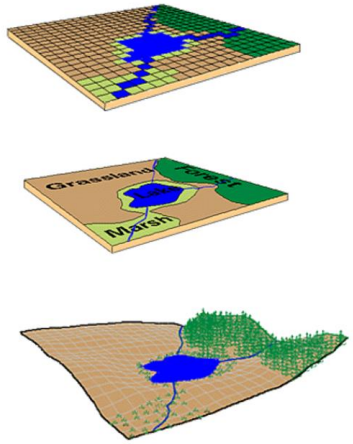
- Aufbau der Datenbank
- Datenverarbeitung
- Datenintegration und -modellierung

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)



- RASTER
- VECTOR
- Real World



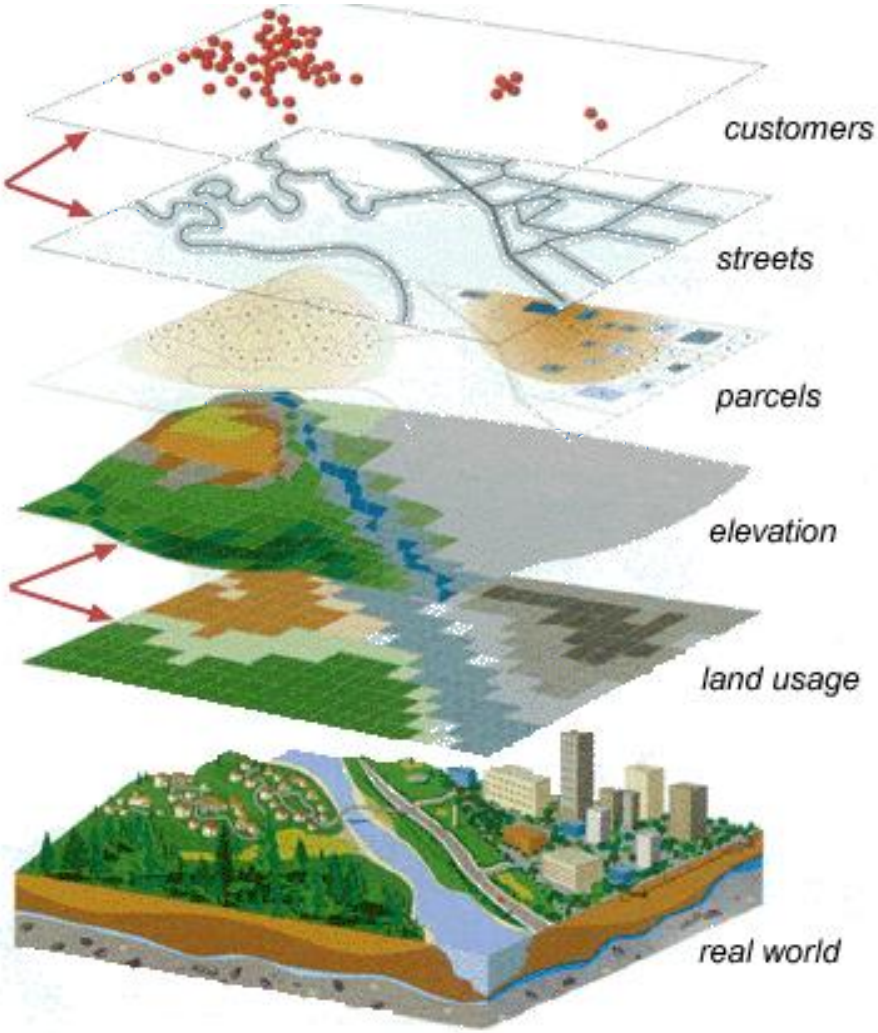
Funktionen eines GIS

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

Grundsätzliche F
- **Dateneingab**
  - Datenb
  - *Datenm*
  - *Datensi*
  - Datenla
    Ebener
    Reihen

v e c t o r

r a s t e r



nd Struktur
nodell)

gruppiert;
rt werden;

Funktionen eines GIS

Ein GIS Projekt un

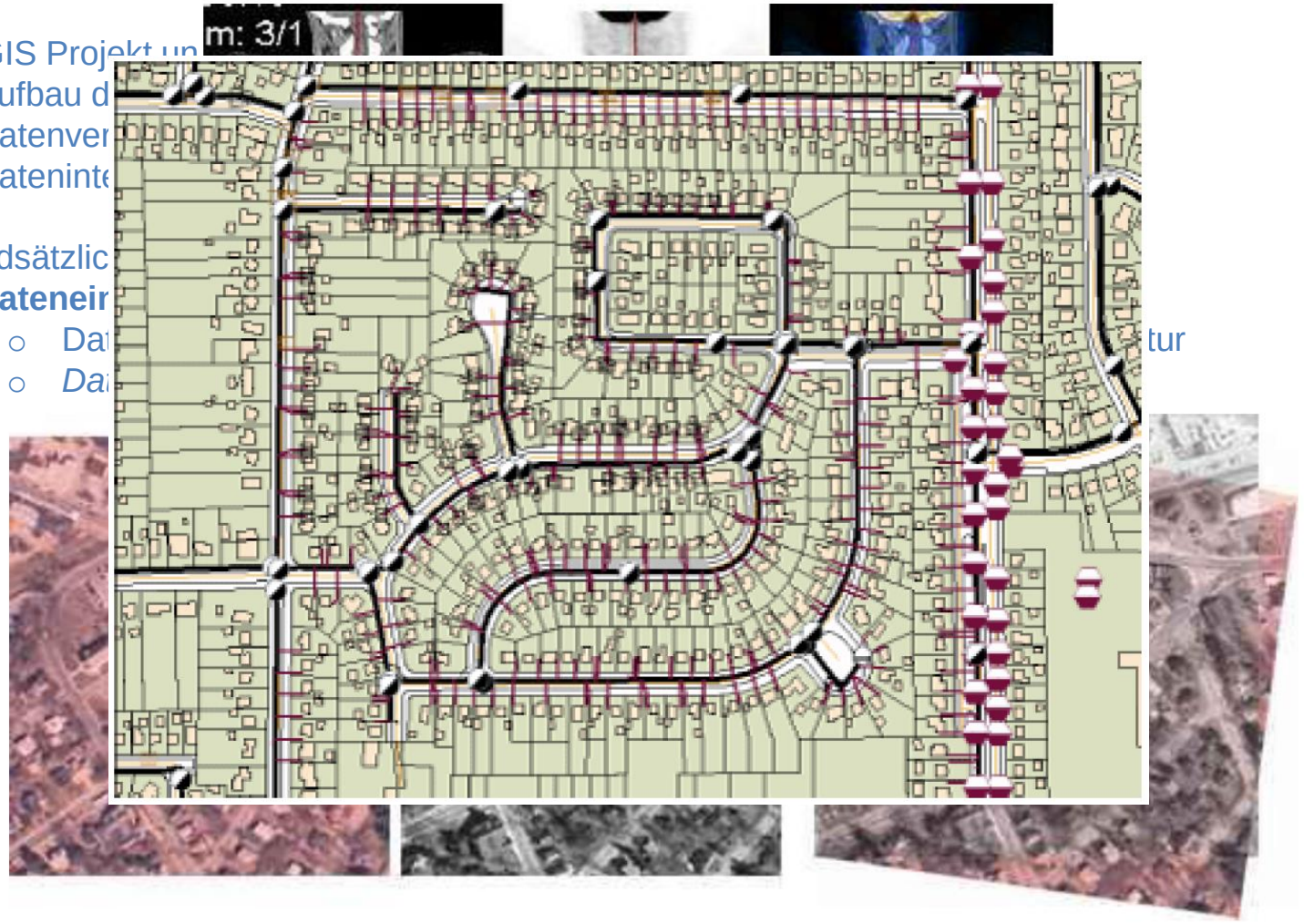
- Aufbau d
- Datenver
- Datenint

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

- **Räumliche A** n als blosse Visualisierur
  - Welch
  - Welch auf?
  - Topolo der Objekten mit wenige

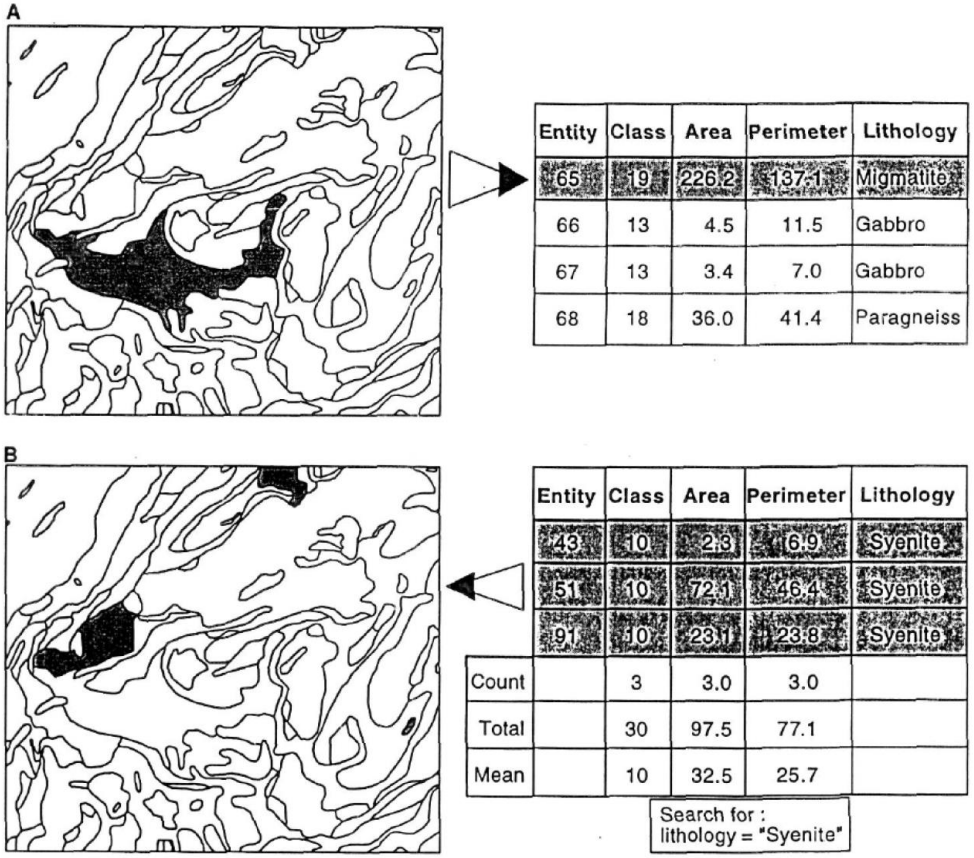


FIG. 5-14. Interactive spatial query of a geological map. **A**. Identifying the attributes of a polygon *selected on a map view*, as indicated in an associated polygon attribute table. **B**. Identifying those polygons on the map that have the attribute called lithology equal to "syenite", as *selected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

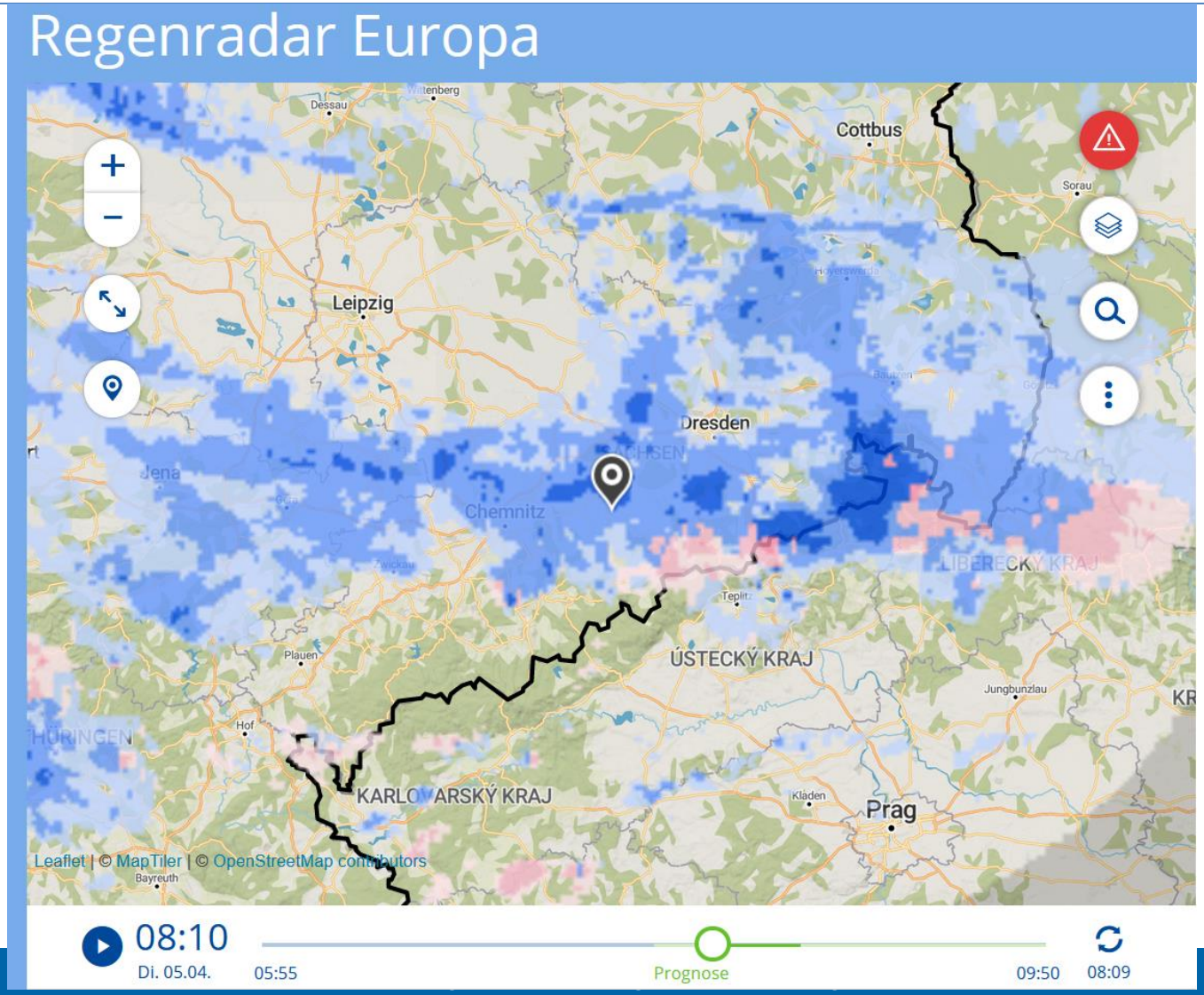
Räumliche Abfrage nach Bonham-Carter (1994)

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)

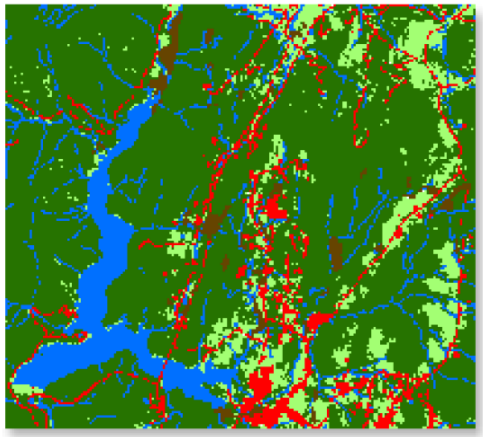
Beispiel: Regenradar

Regenradar


Regenradar Europa
Dessau
Cottbus
Sorau
Leipzig
Dresden
Jena
Chemnitz
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

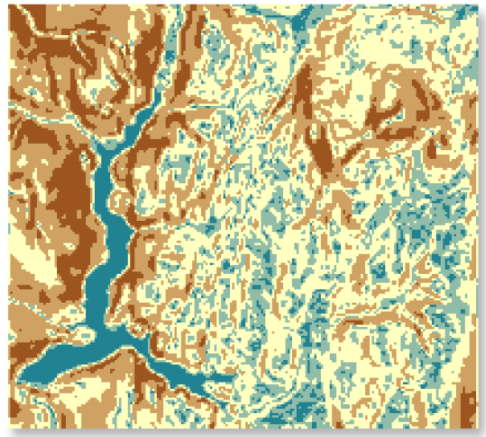
Funktionen eines GIS

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



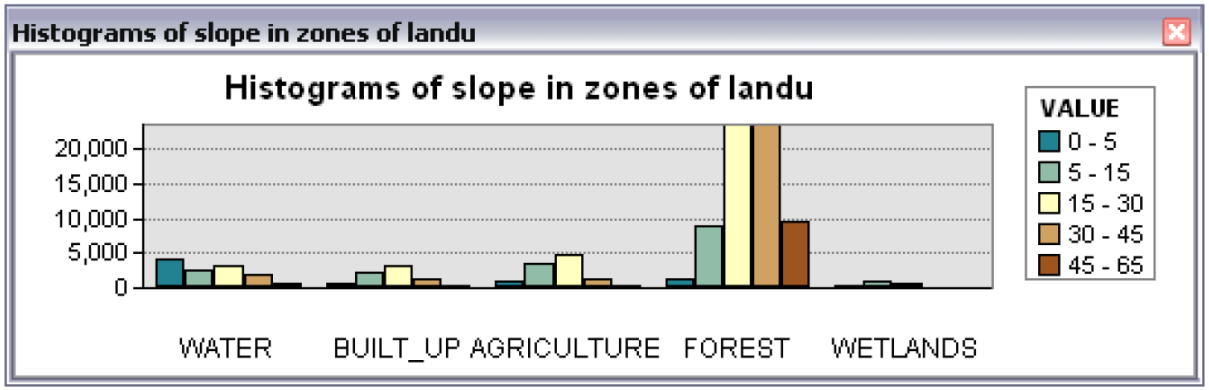


Zone input



Value input

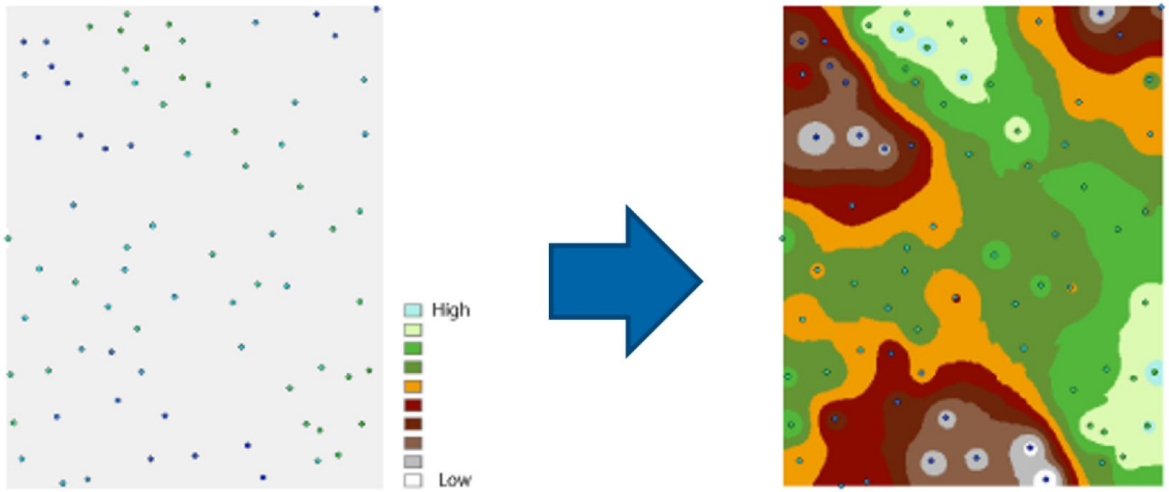
=



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)



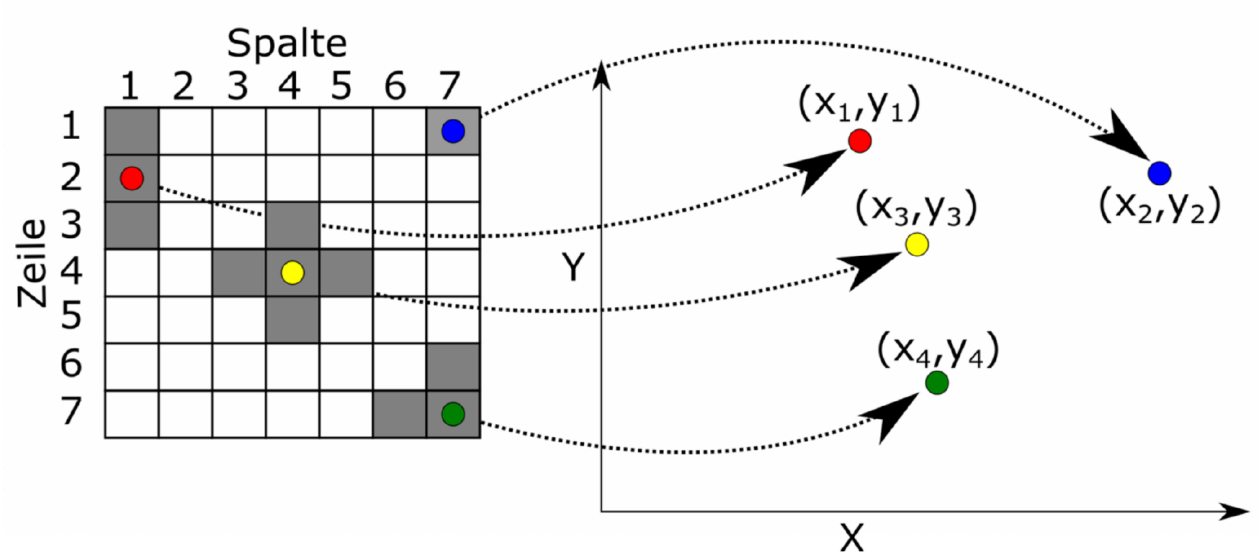
Interpolation

Übersicht

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

Georeferenzierung

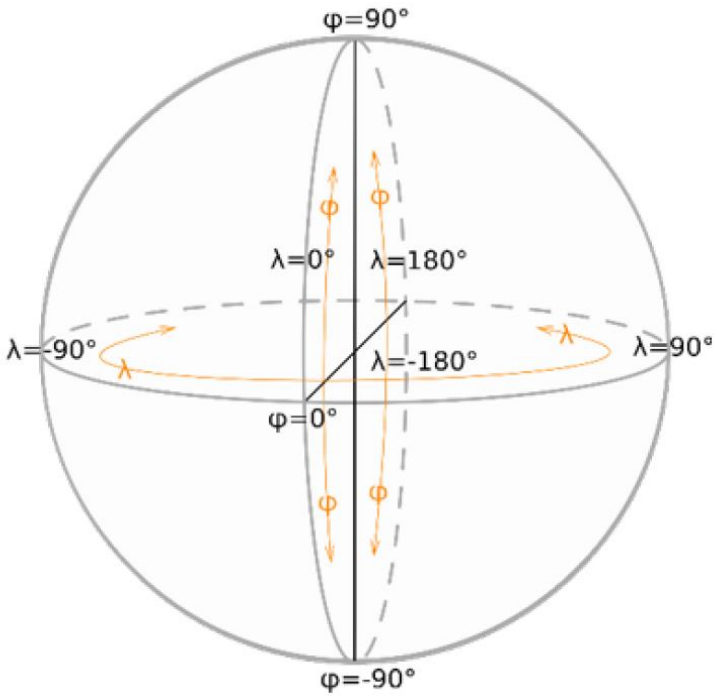
Eine der grundlegenden Aufgaben in einem GIS Projekt ist die Georeferenzierung, um verschiedene Datensätze räumlich miteinander vergleichen und kombinieren zu können (z.B. Luftbildaufnahme, Topographische Karte, geochemische Punktdaten).



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 warden
- Problem 1: eine Karte ist üblicherweise 2-D
- Problem 2: die Erde ist 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.

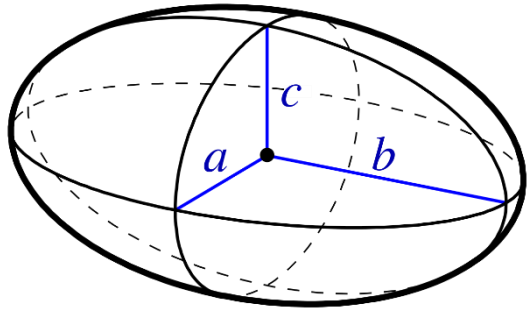
Kartenprojektion

Übliche Referenzflächen:

- Ellipsoid:

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

- **Rotationsellipsoid**: a=b
- Kugel: r=a=b=c (z.B. r=6371.2km)



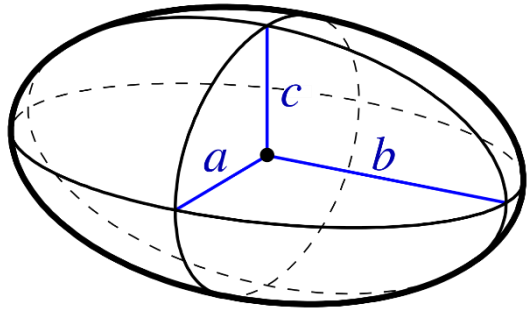
Kartenprojektion

Übliche Referenzflächen:

- Ellipsoid:

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

- **Rotationsellipsoid**: a=b
- Kugel: r=a=b=c (z.B. r=6371.2km)



Wichtige Referenzellipsoide sind z.B.

WGS84 (Nutzung für GPS): a=b=6378.137km, c=6356.752km
Bessel (1841): a=b=6377.379km, c=6356.079km

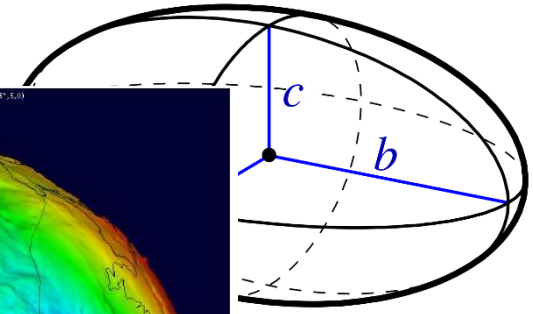
Kartenprojektion

Übliche Referenzflächen:

- Ellipsoid:

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

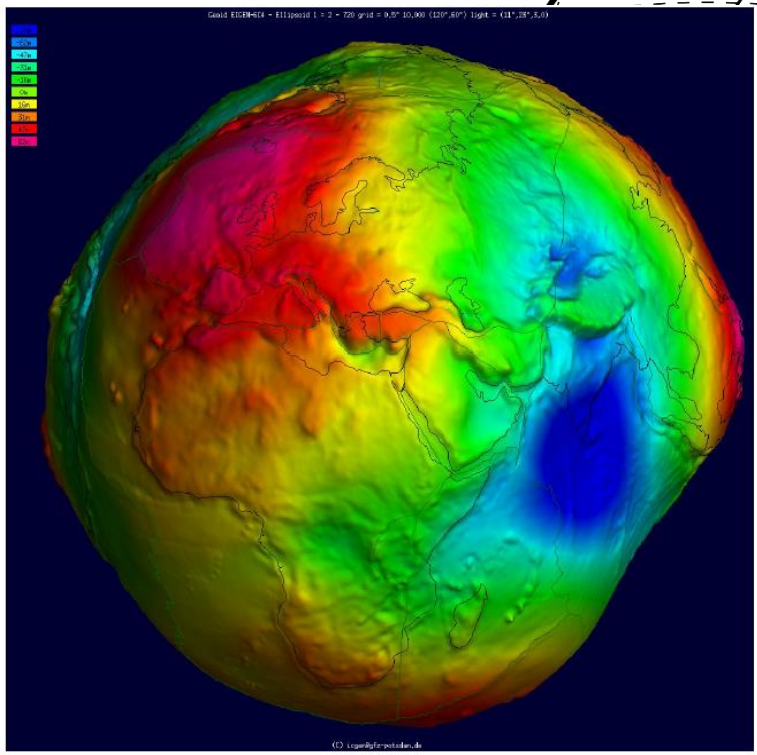
- **Rotationsellipsoid**
- Kugel: r=a=b=c (z.B



Wichtige Referenzellips

WGS84 (Nutzung für G
Bessel (1841): a=b=63

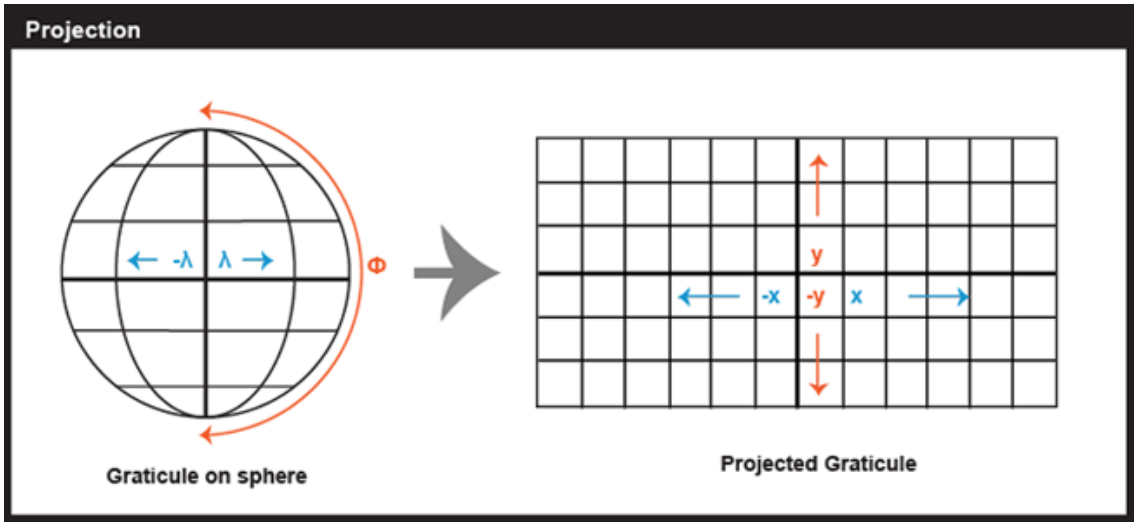
- physikalische mo ächе eines konstanten Gravitationspotentia



icgem.gfz-potsdam.de

Kartenprojektion

Es muss zur Darstellung in 2D jedoch die Oberfläche 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 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 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?

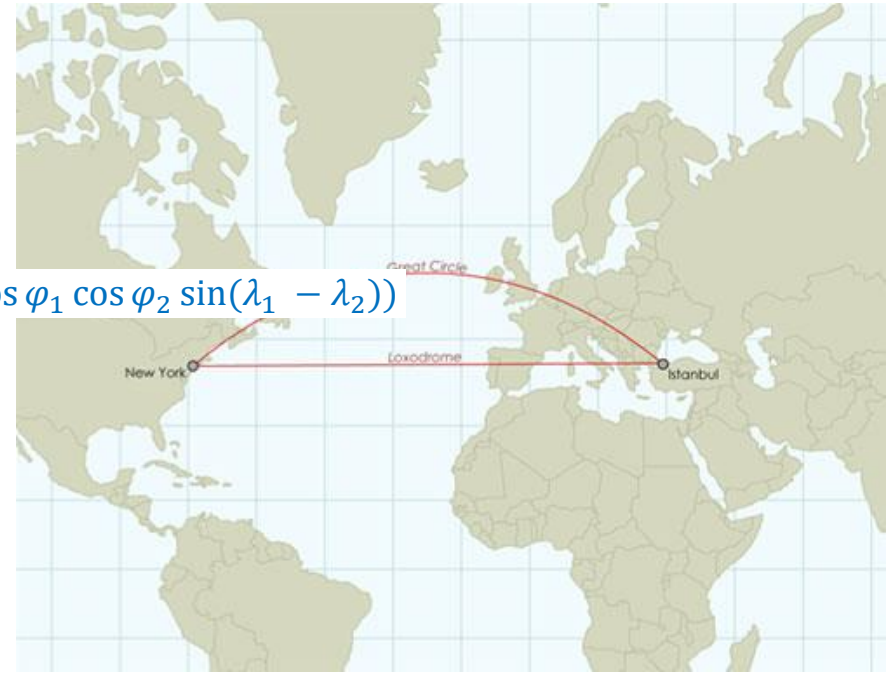
Eine globale verzerrungsfreie Darstellung der Erde ist nicht möglich:

- Winkeltreu
- **Längentreu**
- Flächentreu

Kürzeste Distanz auf kugelförmiger Erde:

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

immer entlang eines Großkreises



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

Kartenprojektion

Es muss zur Darstellung in 2D jedoch die Oberfläche 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:

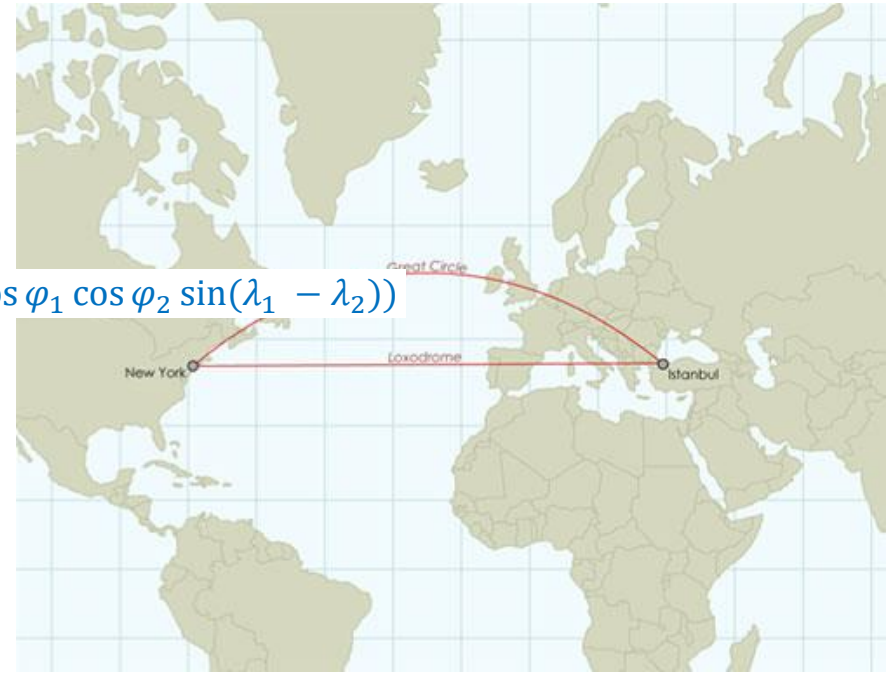
$$\text{dist}_{sph}(P_1, P_2) = r\arcsin(\sin\varphi_1 \sin\varphi_2 + \cos\varphi_1 \cos\varphi_2 \sin(\lambda_1 - \lambda_2))$$

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 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

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

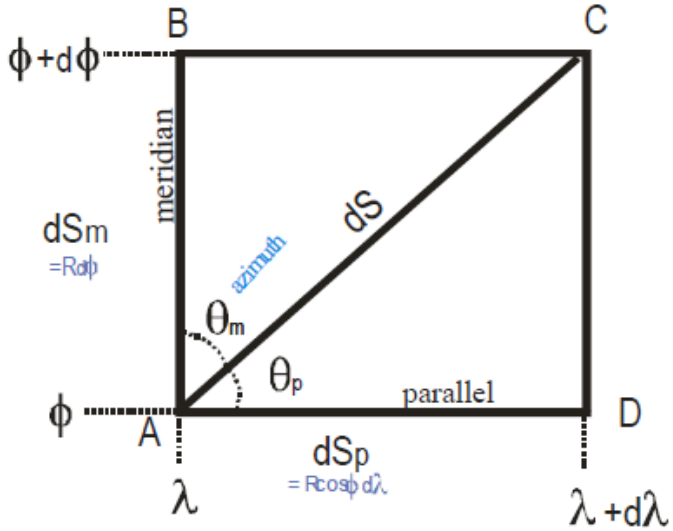
Global längentreue Abbildungen **gibt es nicht**, aber Längentreue unter bestimmten Annahmen kann garantiert warden (z.B. von einem zentralen Punkt aus)



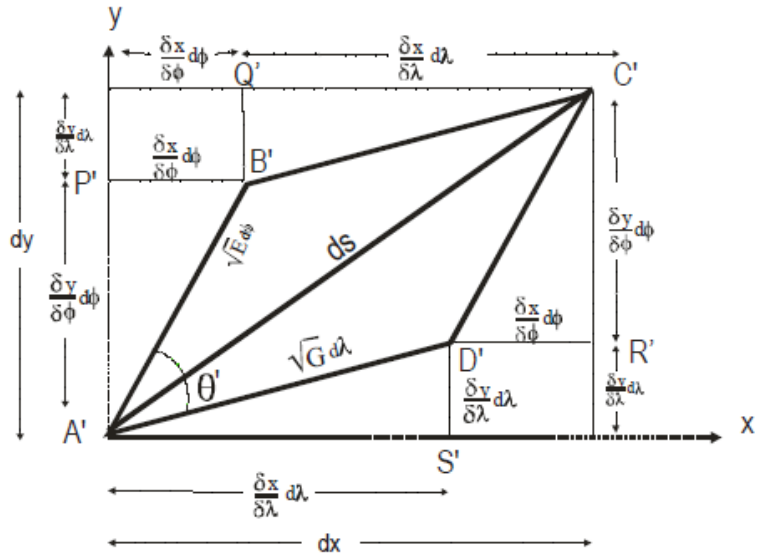
https://www.axismaps.com/guide/map-projections

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

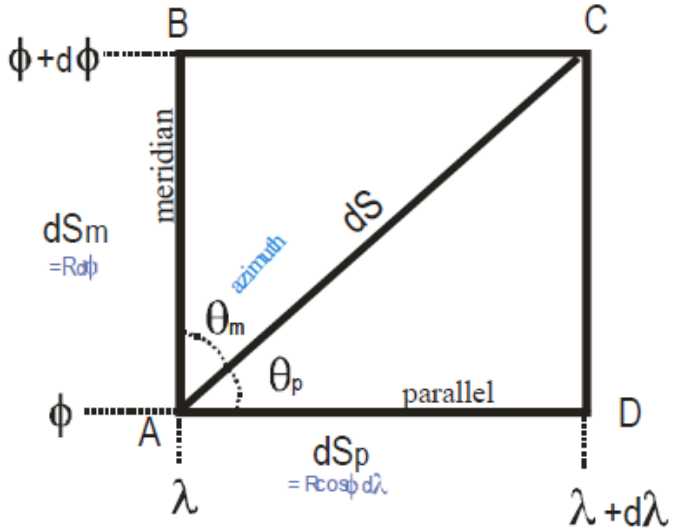
$$E(\,d\varphi\,)^2 = \left(\left(\frac{\partial x(\lambda,\varphi)}{\partial \varphi}\right)^2 + \left(\frac{\partial y(\lambda,\varphi)}{\partial \varphi}\right)^2\right) d\varphi^2$$

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

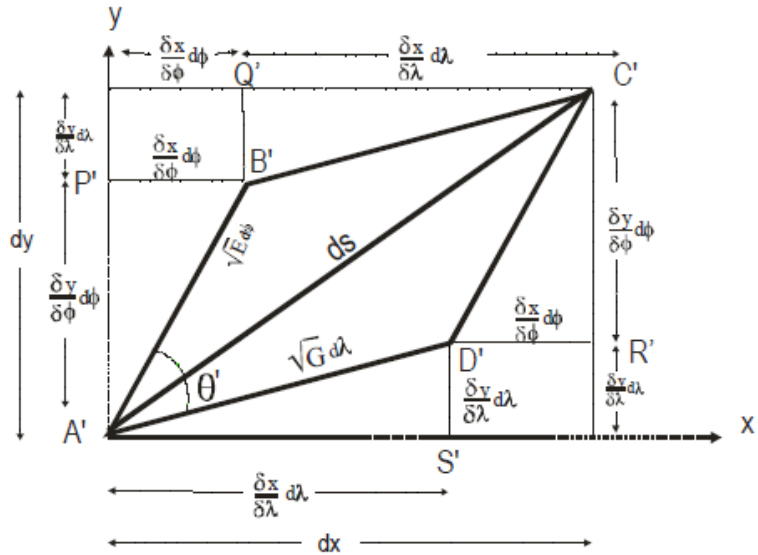
$$F(\,d\varphi d\lambda\,) = \left(\frac{\partial x(\lambda,\varphi)^2}{\partial \varphi \partial \lambda} + \frac{\partial y(\lambda,\varphi)^2}{\partial \varphi \partial \lambda}\right) d\varphi d\lambda$$

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 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?

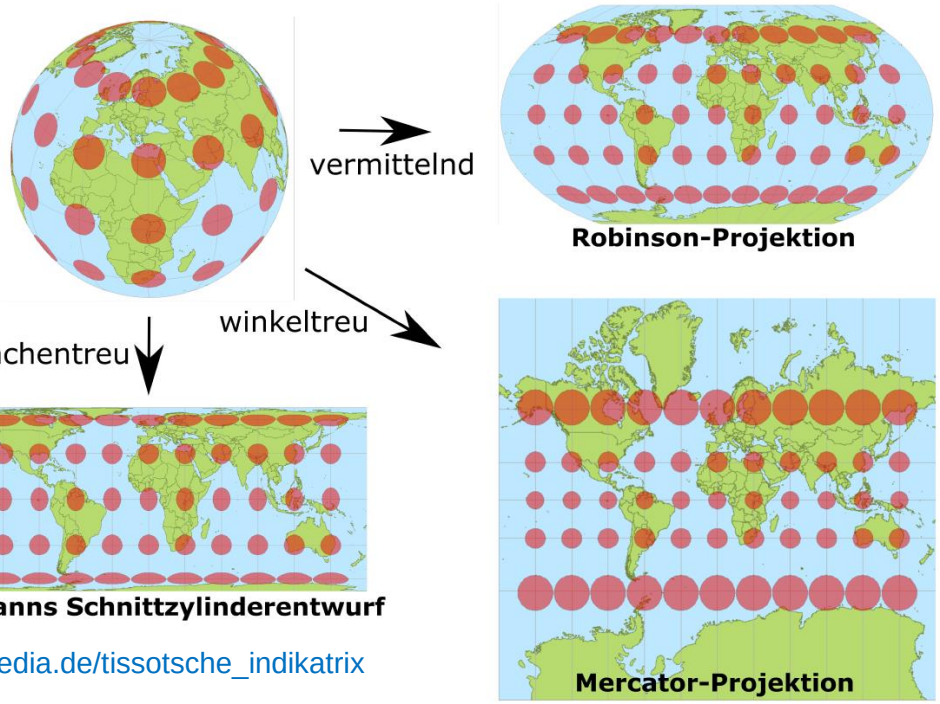
Eine globale verzerrungsfreie Darstellung der Erde ist nicht möglich:

- **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 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?

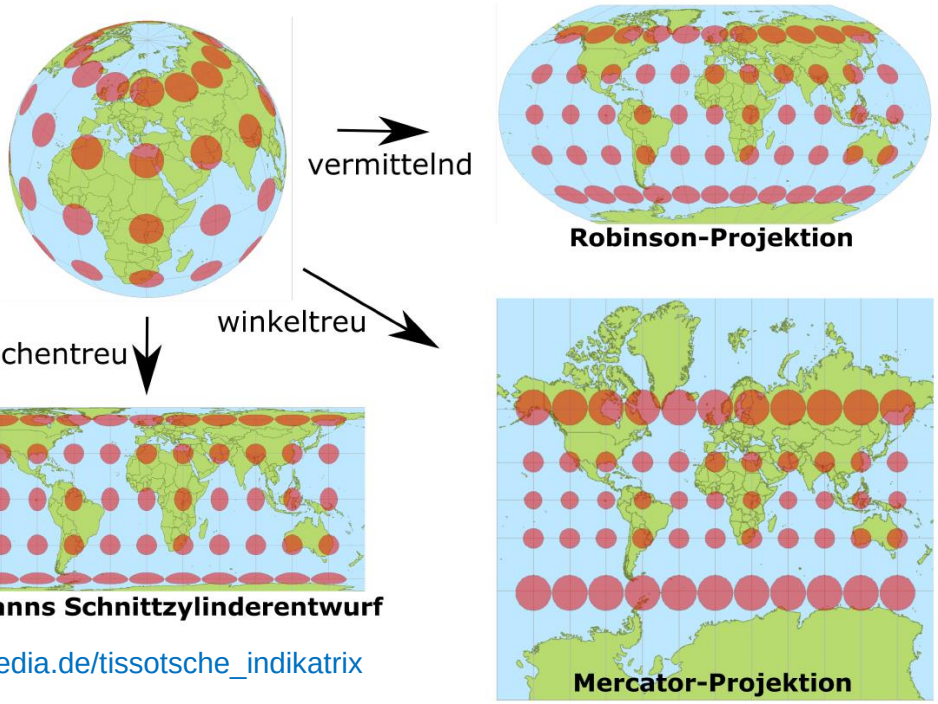
Eine globale verzerrungsfreie Darstellung der Erde ist nicht möglich:

- Winkeltreu
- Längentreu
- **Flächentreu**

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

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

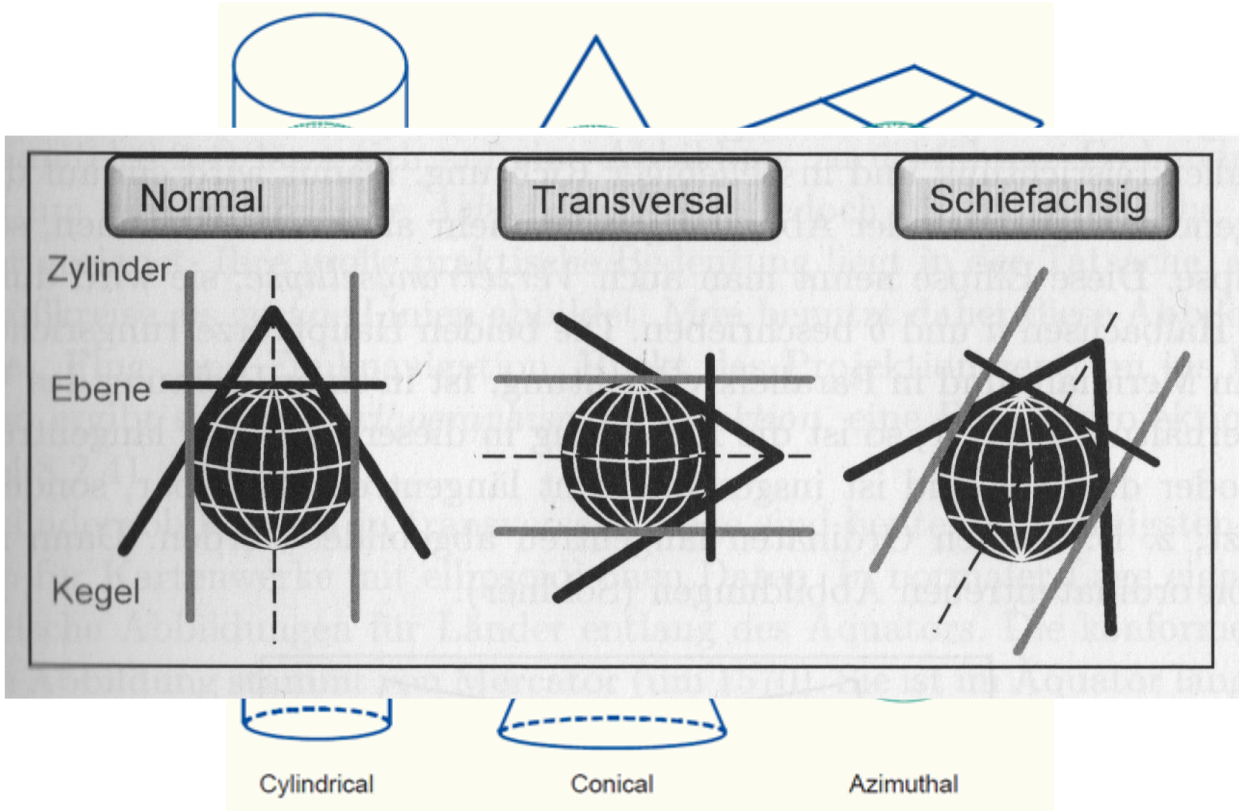
für jedes Gebiet $\Omega \subset \mathbb{S}_r$. Die Konstante α spiegelt dabei den Massstab wieder (Massstab sei hier als ein Skalierungsfaktor aufgefasst, nicht der umgangs-sprachliche Massstab).



wikipedia.de/tissotsche_indikatrix

Kartenprojektion

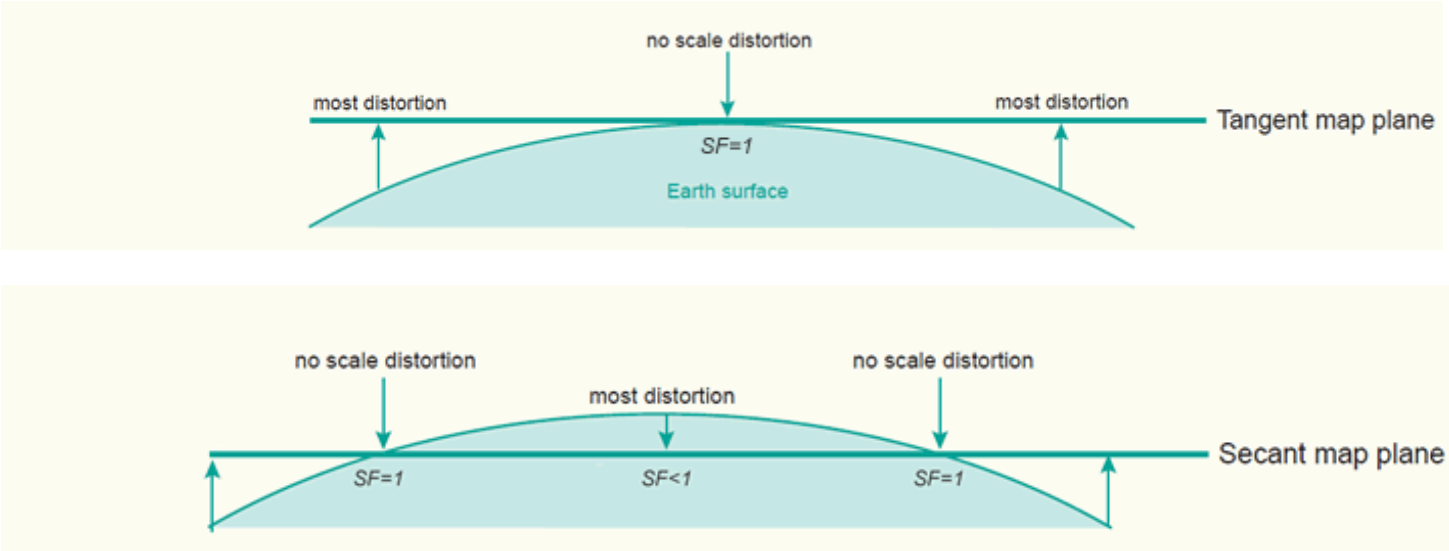
Man unterscheidet Kartenprojektionen basierend auf den Projektionsflächen



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:

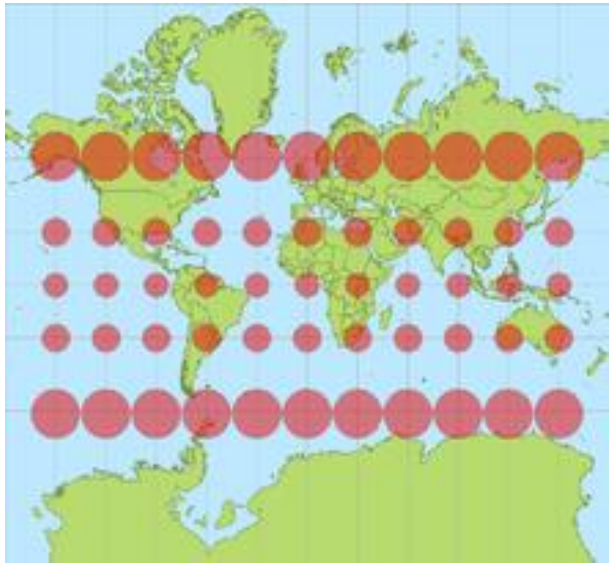
- **Mercator Projektion** (Zylinder-Projektion)

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

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

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

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



Kartenprojektion

Beispiele für winkeltreue Projektionen:

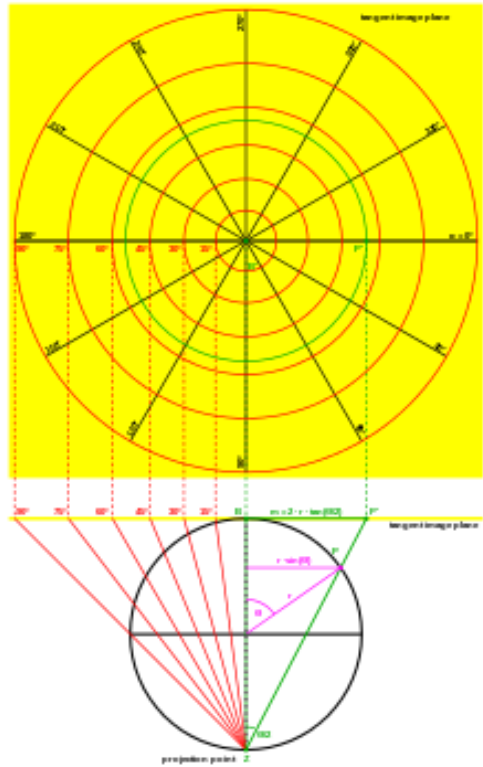
- **Mercator Projektion** (Zylinder-Projektion)

  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

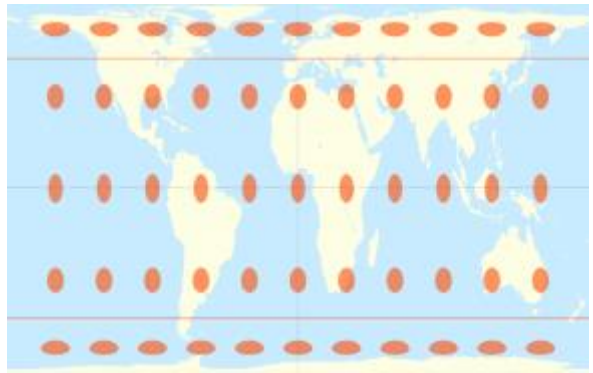
Beispiele für flächentreue Projektionen:

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

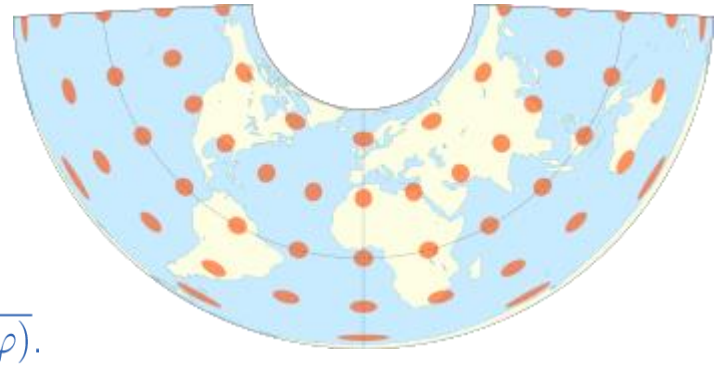
  $$x = r\lambda$$
  $$y = 2r \sin(\varphi)$$

- **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 $\rho = \frac{r}{\beta}\sqrt{\gamma - 2\beta \sin(\varphi)}$.

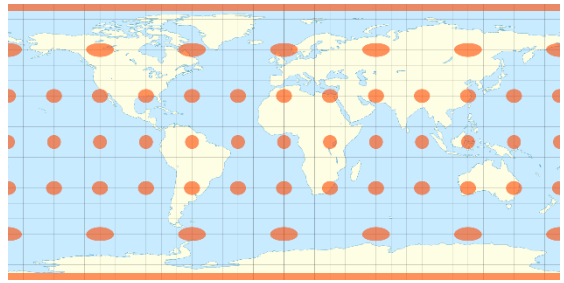


Kartenprojektion

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:

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



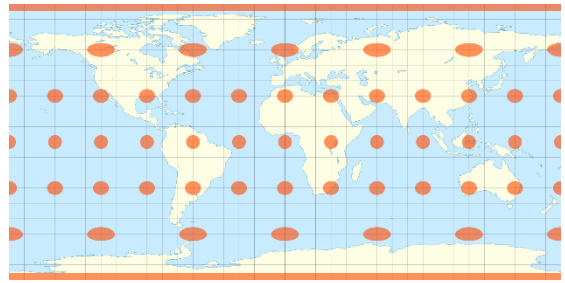
https://de.wikipedia.org/wiki/Plattkarte

Kartenprojektion

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

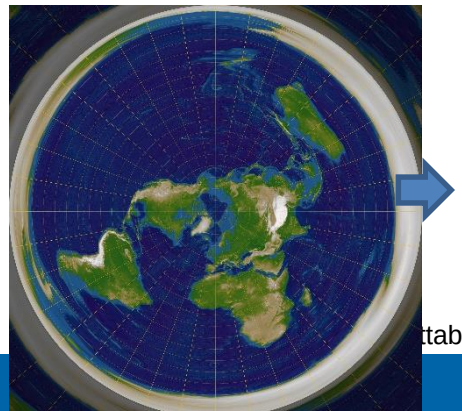
- **Quadratischde Plattkarte**: längentreu entlang der Meridiane
  - Direkte Abbildung der geografischen Koordinaten in die Ebene
  - Grundidee: $x = \lambda$ und $y = \varphi$
  - Transformation:

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



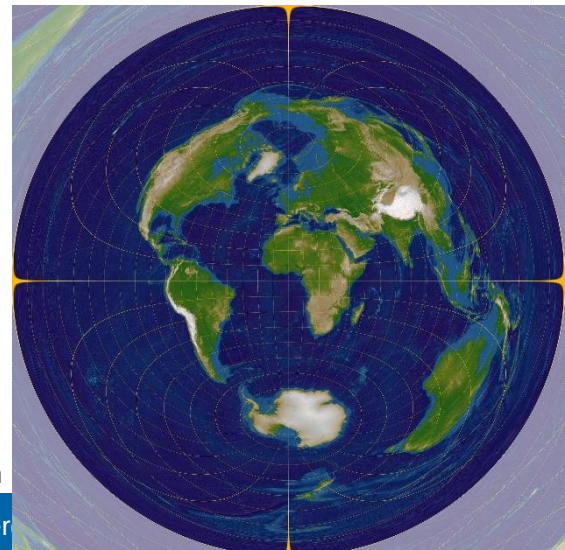
https://de.wikipedia.org/wiki/Plattkarte

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





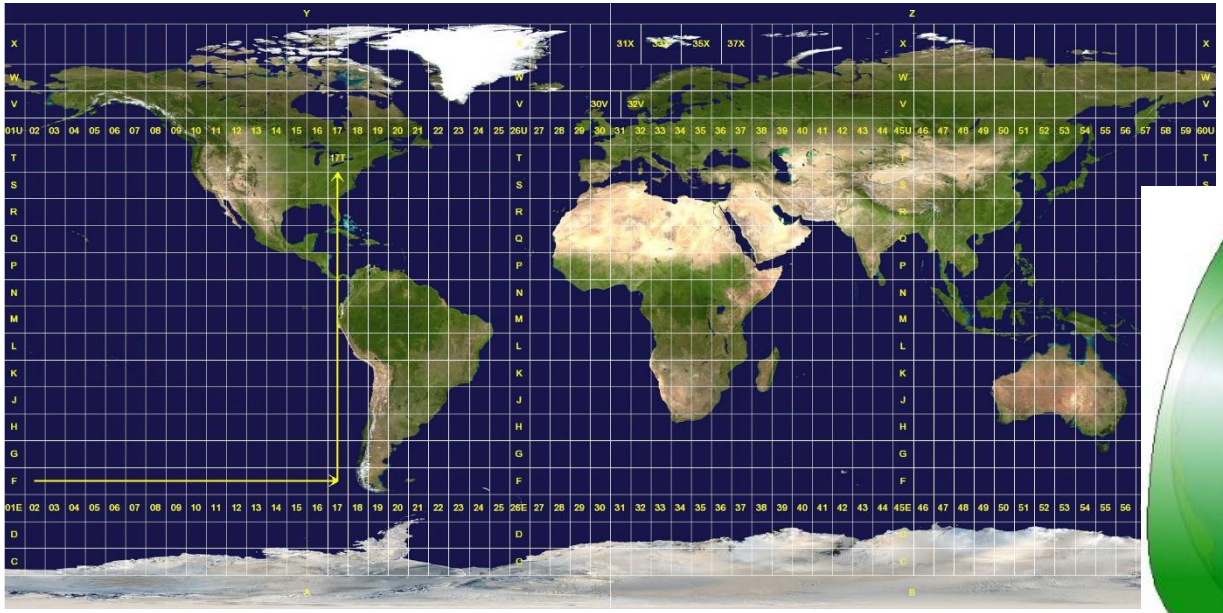
ttabstandstreue_Azimutalprojektion



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)

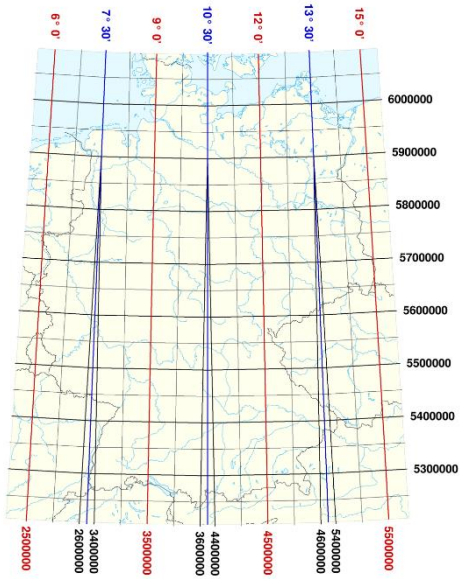




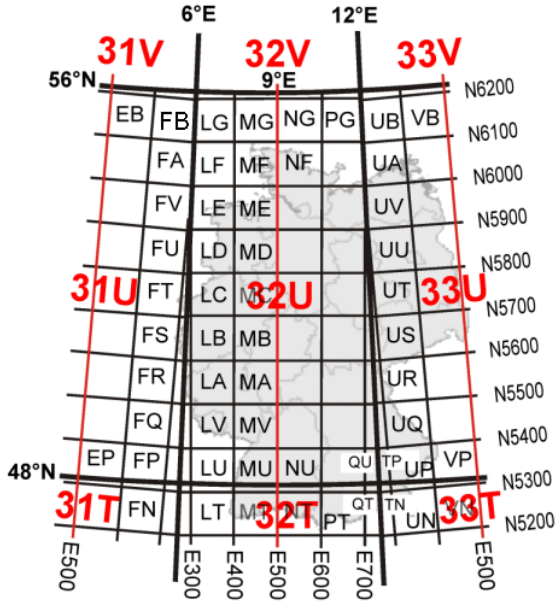
Kartenprojektion

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

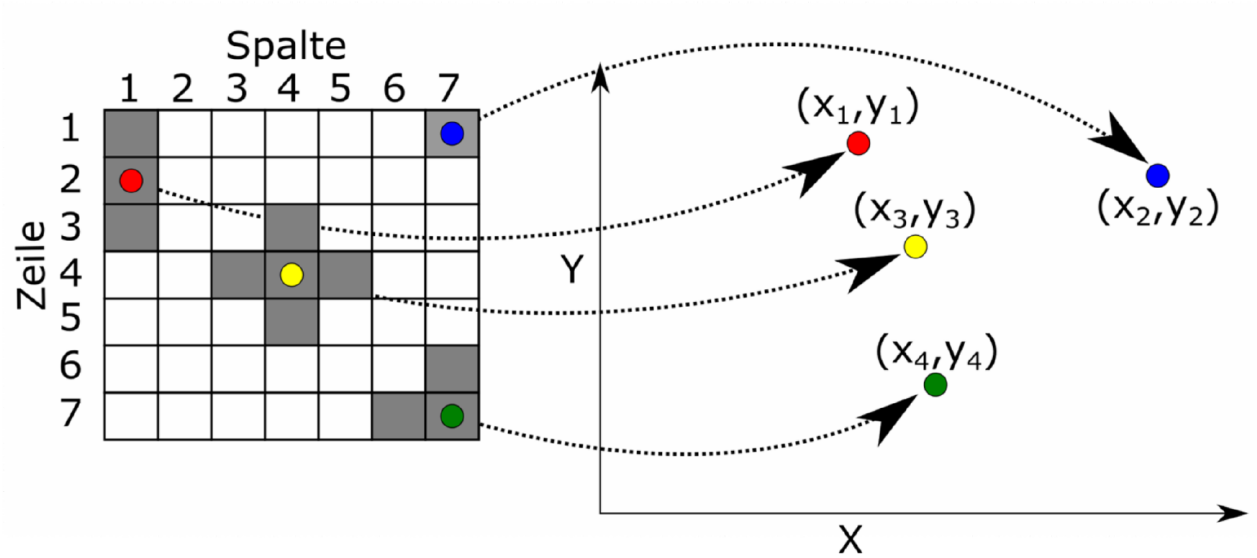


UTM

Georeferenzierung

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).

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



Koordinatenumwandlung

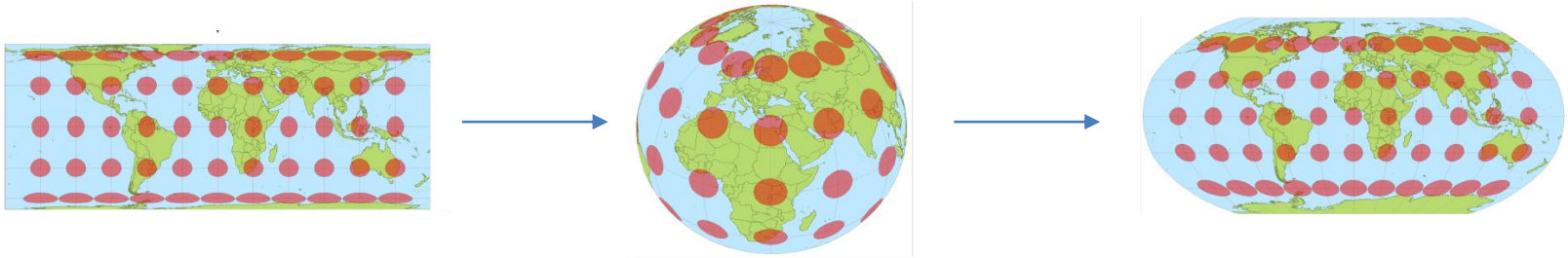
Georeferenzierung umfasst (grob gesagt) den Workflow, welcher den externen Raumbezug der eigentlichen Daten in den Raumbezug des GIS-Projektes umwandelt. Es muss der externe Raumbezug (z.B. geographische Koordinaten (λ,φ) oder Pixel in einem Luftbild) in ein üblicherweise durch (x,y)-Koordinaten gegebenen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

Wir unterscheiden oft zwei Koordinatenoperation:

- **Koordinatenumwandlung**
- Koordinatentransformartion

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

wobei T_1 und T_2 z.B. Kartenprojektionen aus der vorherigen Vorlesung sind.



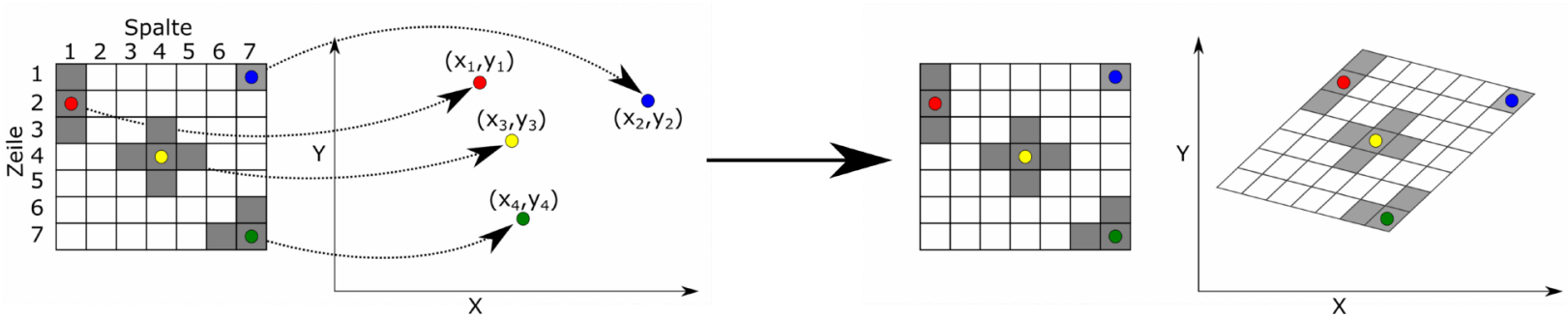
Koordinatentransformation

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
- **Koordinatentransformartion**

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

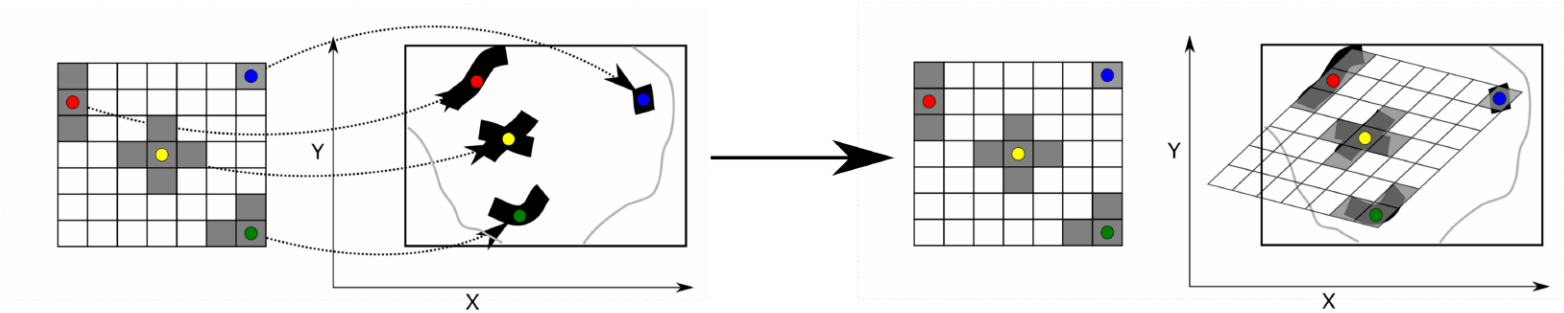
Koordinatentransformation

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
- **Koordinatentransformartion**

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

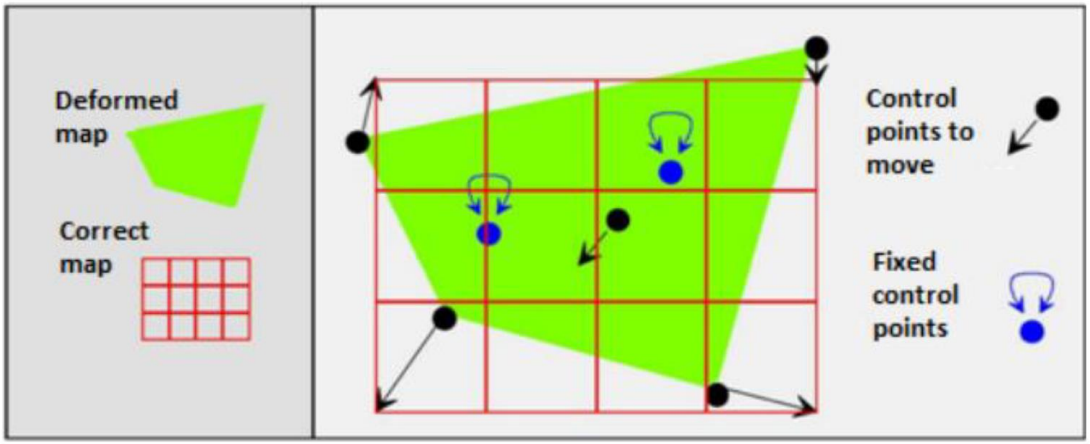
Affin-lineare Abbildungen können zwischen lokalen Koordinaten (x,y) und (x',y') in zwei verschiedenen Koordinatensystemen transformieren.

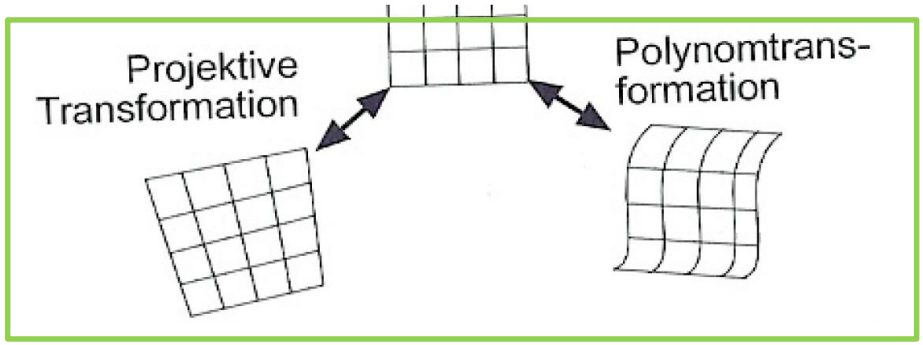
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{M}$$

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** bekannte, 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** bekannte, 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** bekannte, 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** bekannte, 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

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

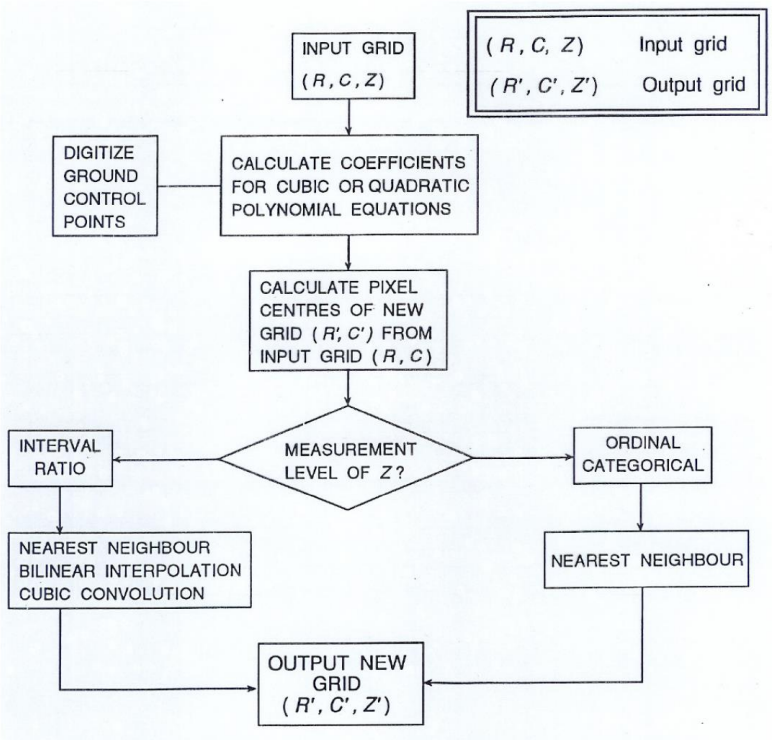
wobei $m_{1,1}, \ldots, m_{2,2}$ die (unbekannten) Einträge der Matrix $\mathbf{M}$ sind und v_1, v_2 die (unbekannten) Einträge des Vektors $\mathbf{v}$ sind.

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)

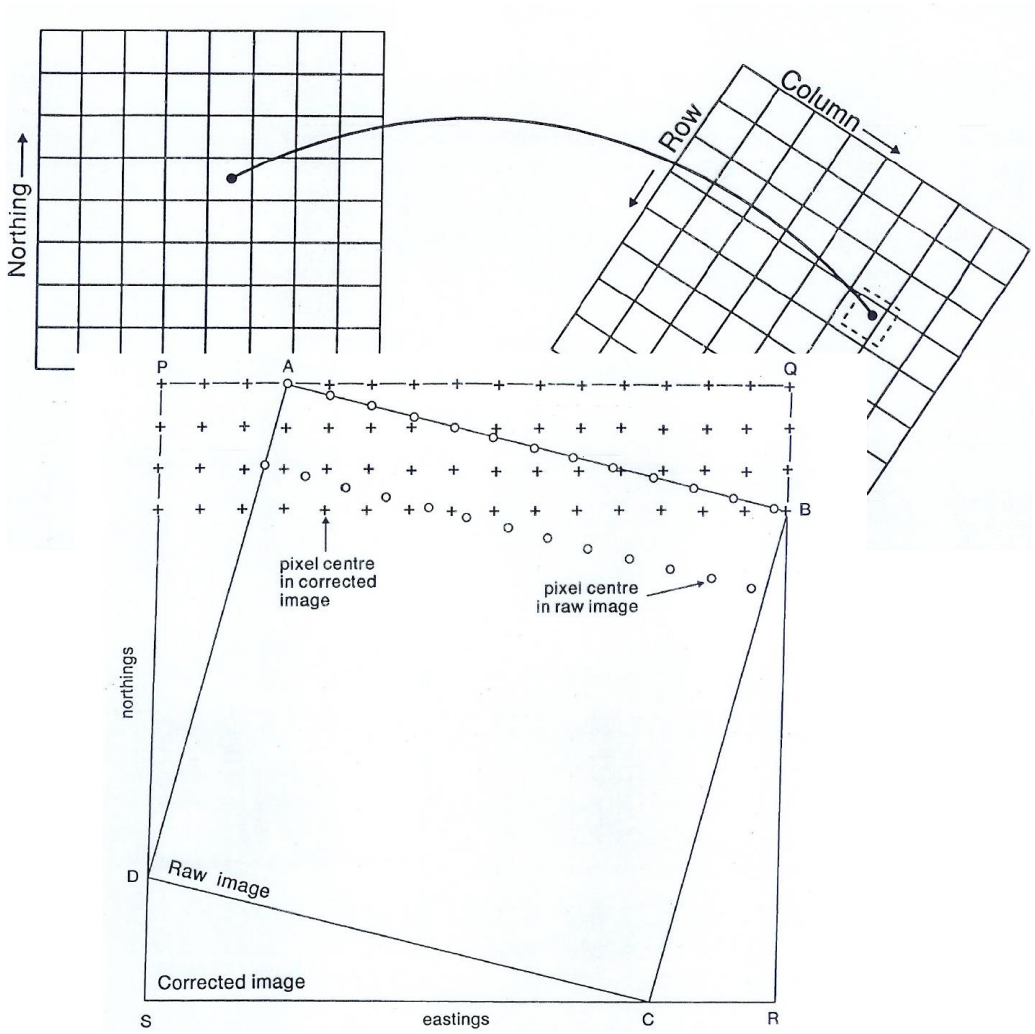
Georeferenzierung von Rasterdaten

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.



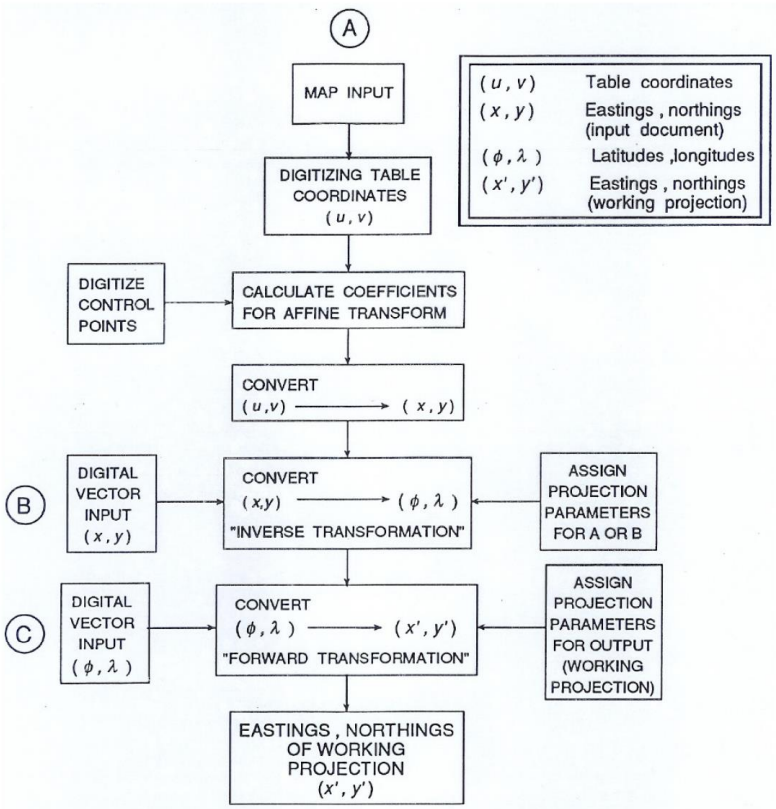
Bonham-Carter: Geographic Information Systems for Geoscientists: Modeling with GIS (1994)

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”

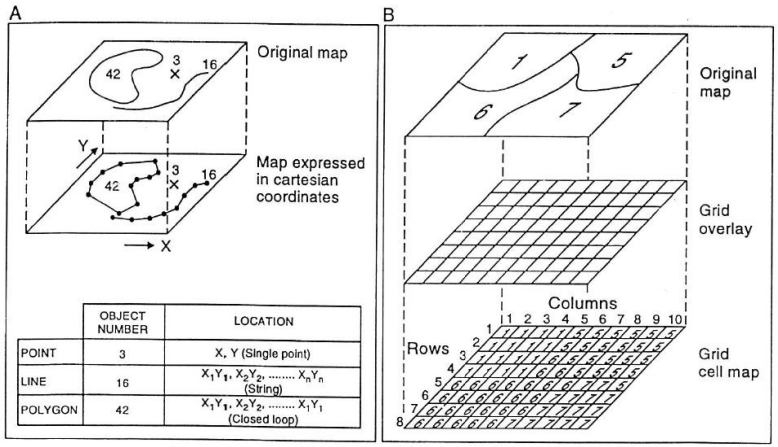
Objekttypen

- **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



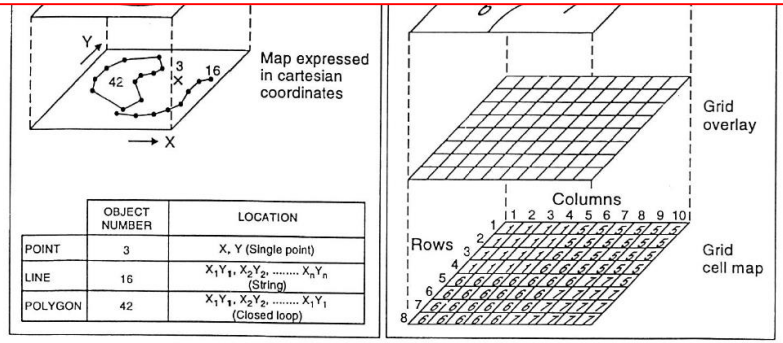
	OBJECT NUMBER	LOCATION
POINT	3	X, Y (Single point)
LINE	16	$X_1Y_1, X_2Y_2, \ldots\ldots X_nY_n$ (String)
POLYGON	42	$X_1Y_1, X_2Y_2, \ldots\ldots X_1Y_1$ (Closed loop)

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

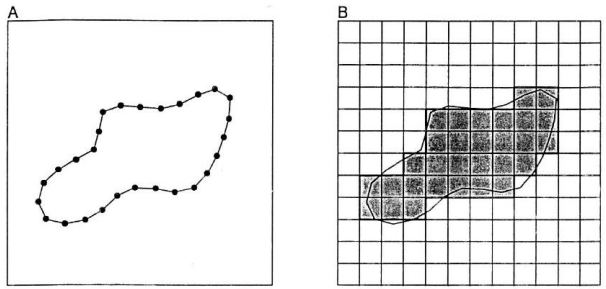
- **Vektormodell**:
  - Grundelegende Einheiten sind Vertexe (0-D Punktobjekte); und darauf aufbauend Linien/Polygonzüge, Polygone, …
  - Exakte Koordinaten für Vertexe, kein Auflösungsverlust
  - jedem Element kann ein oder mehrere Attributwerte zugeordnet werden

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



Raster- vs. Vektormodell

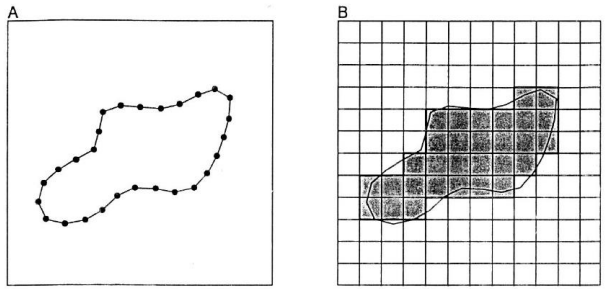
- **Rastermodell**:
  - ausschliesslich flächenhafte Betrachtung von Objekten: jedes Pixel beschreibt eine (möglicherweise sehr kleine) Fläche
  - größere Einzelobjekte lassen sich nur über Attributwerte voneinander abgrenzen
  - Datenerfassung einfach und effizient
  - Position und Nachbarschaftsbeziehungen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

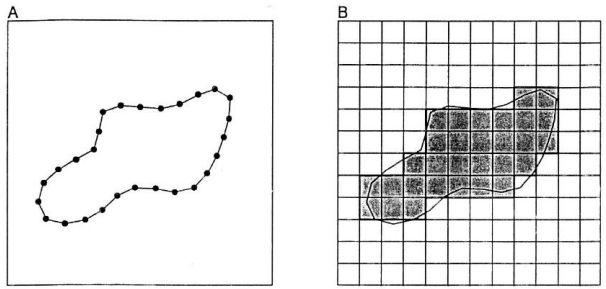
Bestimmung von Umfang und Fläche eines Objektes

- **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:

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

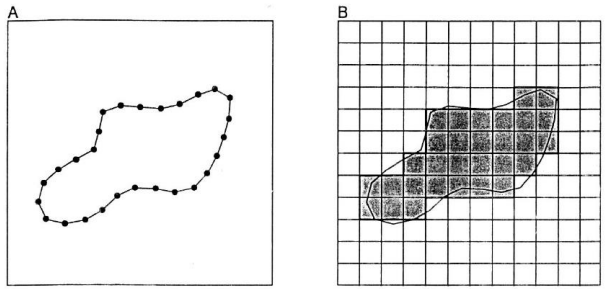


Raster- vs. Vektormodell

Bestimmung von Umfang und Fläche eines Objektes

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

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

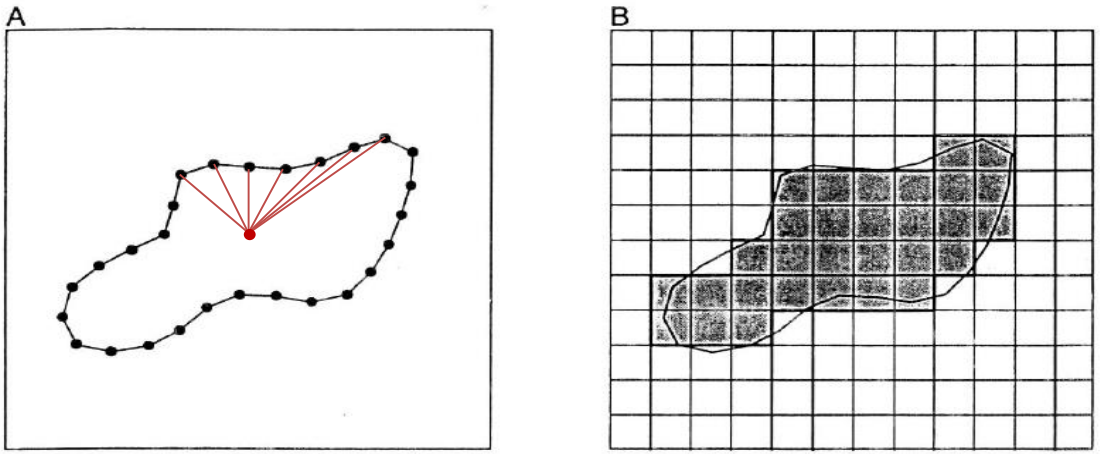


Raster- vs. Vektormodell

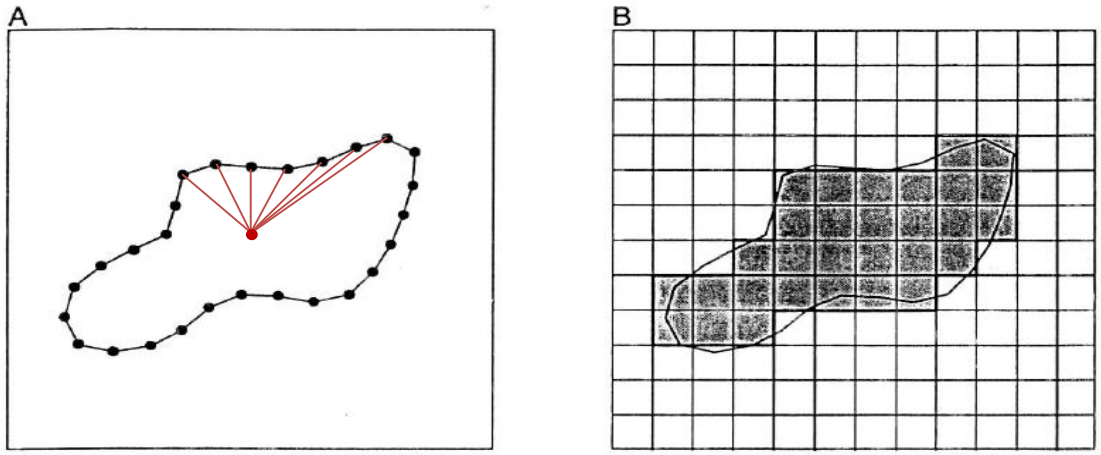
Bestimmung von Umfang und Fläche eines Objektes

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

- **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.

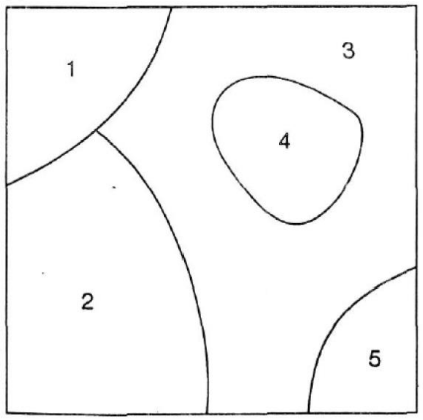
Raster- vs. Vektormodell

Geoobjekte können über verschiedene Attribute beschrieben warden:

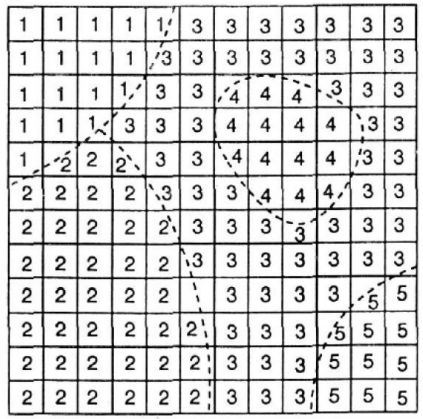
- räumliche Attribute
- zeitliche Attribute
- thematische A

Die Attribute k hinterlegt werden. Möglichkeit der \

A



B



C

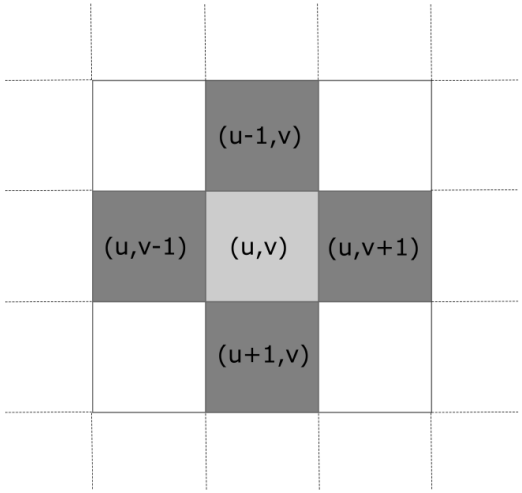
Polygon	Class	Rock type	Age	Name
1	15	sandstone	Late Pennsylvanian	Andrews Formation
2	6	limestone	Early Silurian	Barry Formation
3	3	shale	Middle Silurian	Clinton shale
4	14	granite	Devonian	Delta granite
5	14	granite	Devonian	Delta granite

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

Rastermodell

Rasterdaten können sehr einfach als Matrix $A \in \mathbb{R}^{n \times n}$ 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 $A \in \mathbb{R}^{n \times n}$ 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)

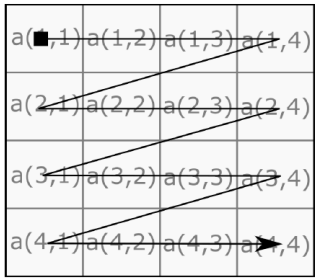
Band A

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)

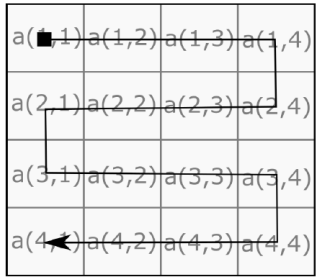
Band B

Rastermodell

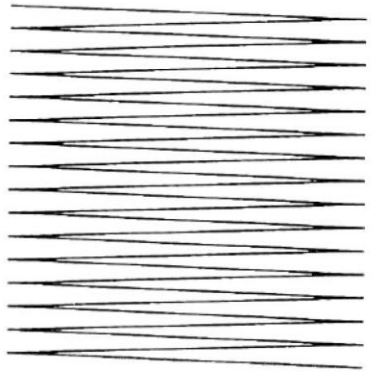
Oft ist es effizient und einfach Matrixeinträge sequentiell in einer Zeile abzulegen. Dann muss jedoch die Durchlaufvorschrift der Matrix bekannt sein!

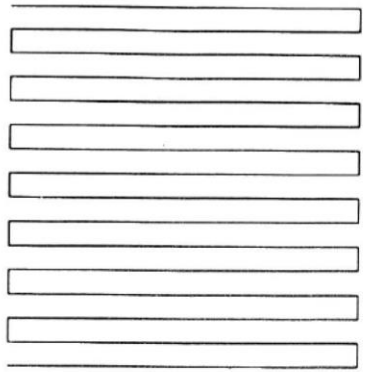


Band A



Band A

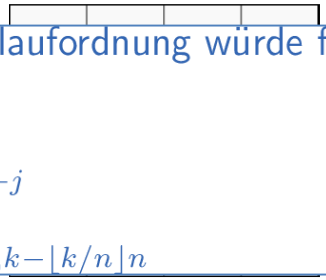

(a) Row Order

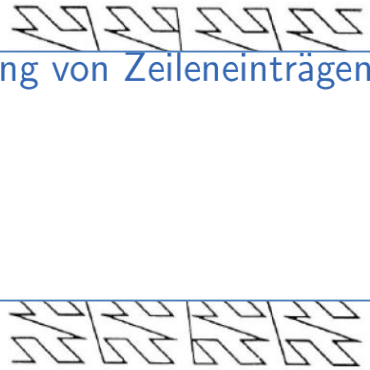

(b) Row-prime Order

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

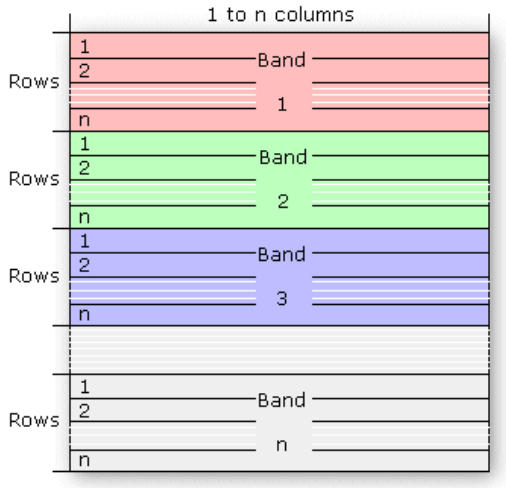

(d) Pi-Order

Rastermodell

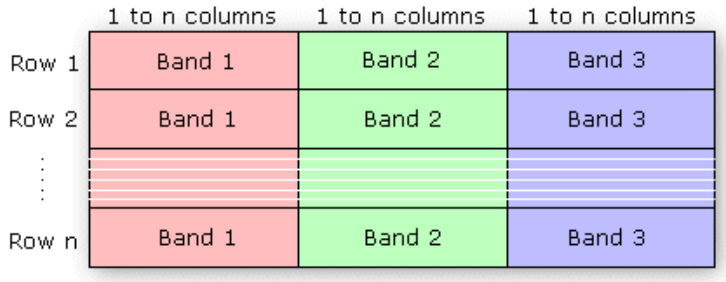
Angenommen wir wollen mehrere Bänder/Layer durchlaufen (z.B. bei hyperspektralen Daten). Welche Durchlaufvorschriften gibt es dann?

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

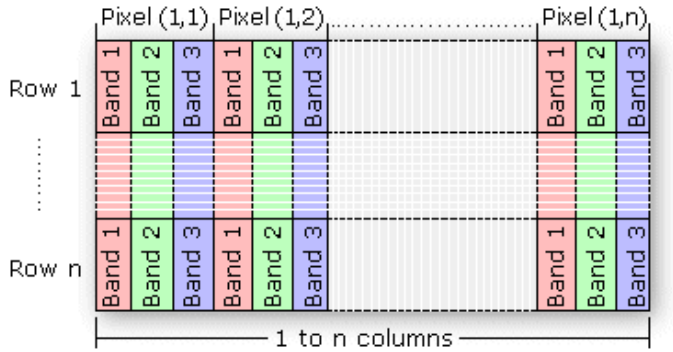
Band sequential (BSP)



Band Interleaved by Line (BIL)



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) ... 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), …

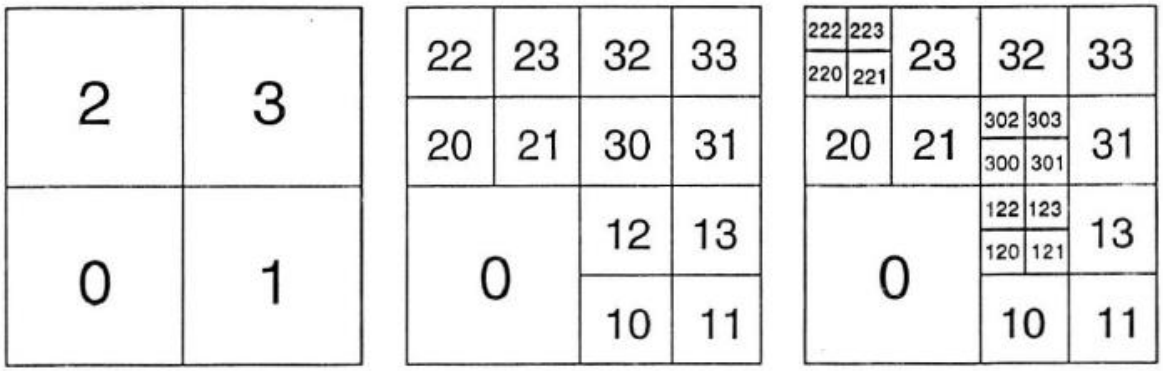
Die ersten zwei Einträge beschreiben die Zeile und Spalte der linken oberen Ecke eines Quadrates, der dritte Eintrag die Seitenlänge des Quadrates, der vierte Eintrag das Attribut.

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

Rastermodell

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

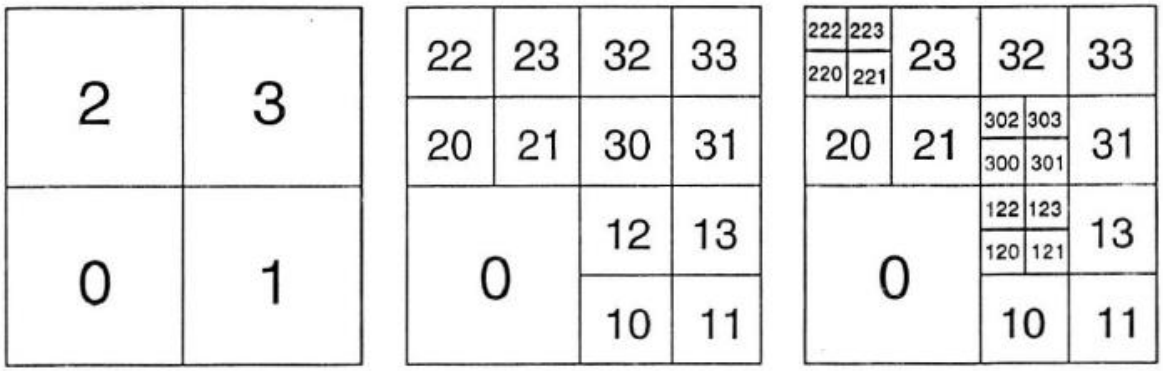
Ein weiteres auf flächige homogene Objekte abzielendes System. Sukzessive, hierarchische Unterteilung des Rasters in kleiner werdende Quadranten.



Rastermodell

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

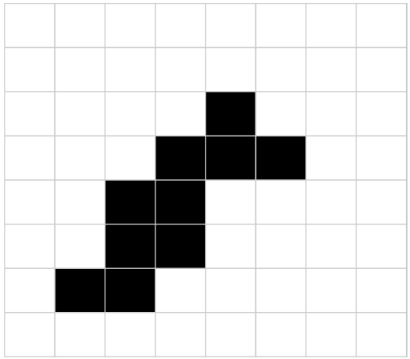
Ein weiteres auf flächige homogene Objekte abzielendes System. Sukzessive, hierarchische Unterteilung des Rasters in kleiner werdende Quadranten.

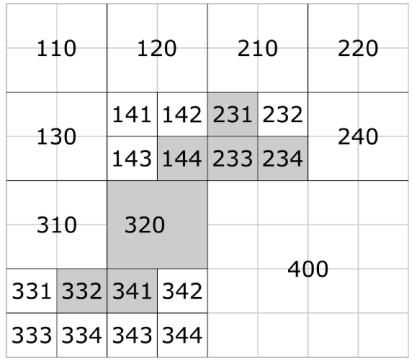


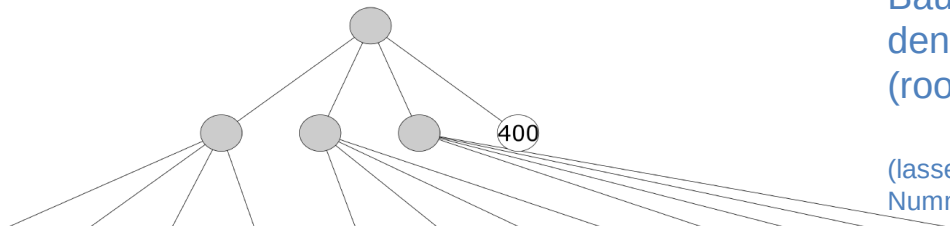
Es bietet sich an die Quadranten mit einem Zahlsystem zur Basis vier zu versehen, für eine intuitive Nummerierung der Quadranten.

Rastermodell

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







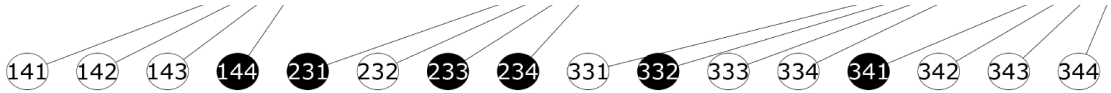
Die Zerlegung des linken Rasters mittels Quadtrees sähe wie folgt aus:

(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)

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

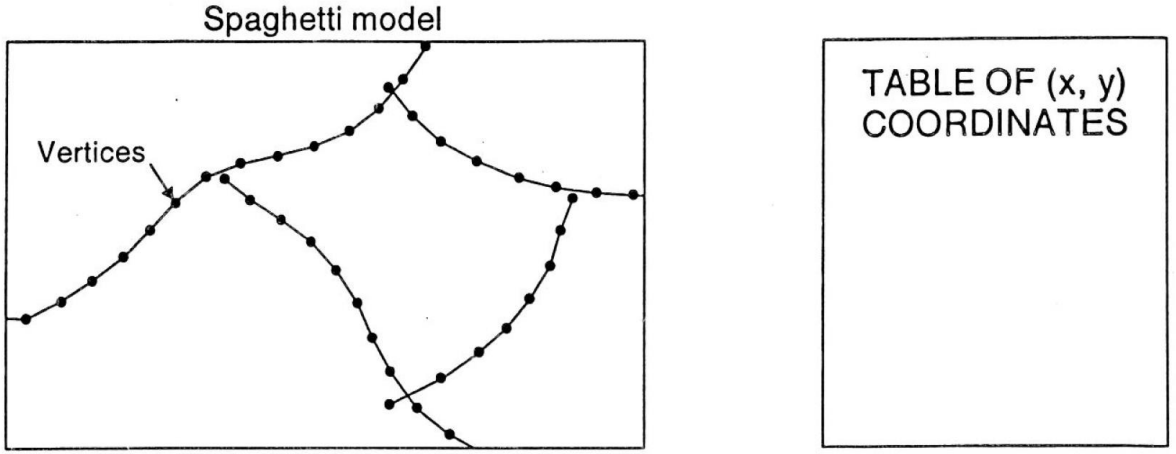


Wie verhält sich Speicheraufwand von Quadtrees zur vorher beschriebenen Blockkodierung?

Vektormodell

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

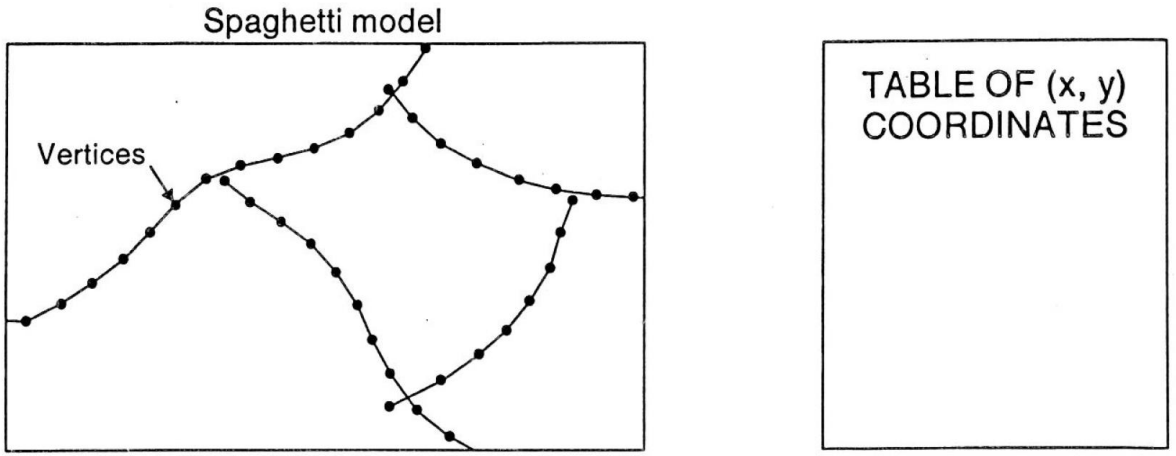
Die einfachste Datenstruktur ist die **Sphagettistruktur**:



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}

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.

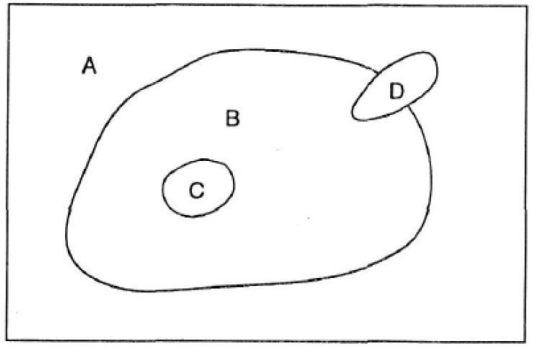
Einzelne Objekte werden als bloss ... (Reihenfolge spielt eine Rolle). ... übereinstimmen.

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 notwedig.

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

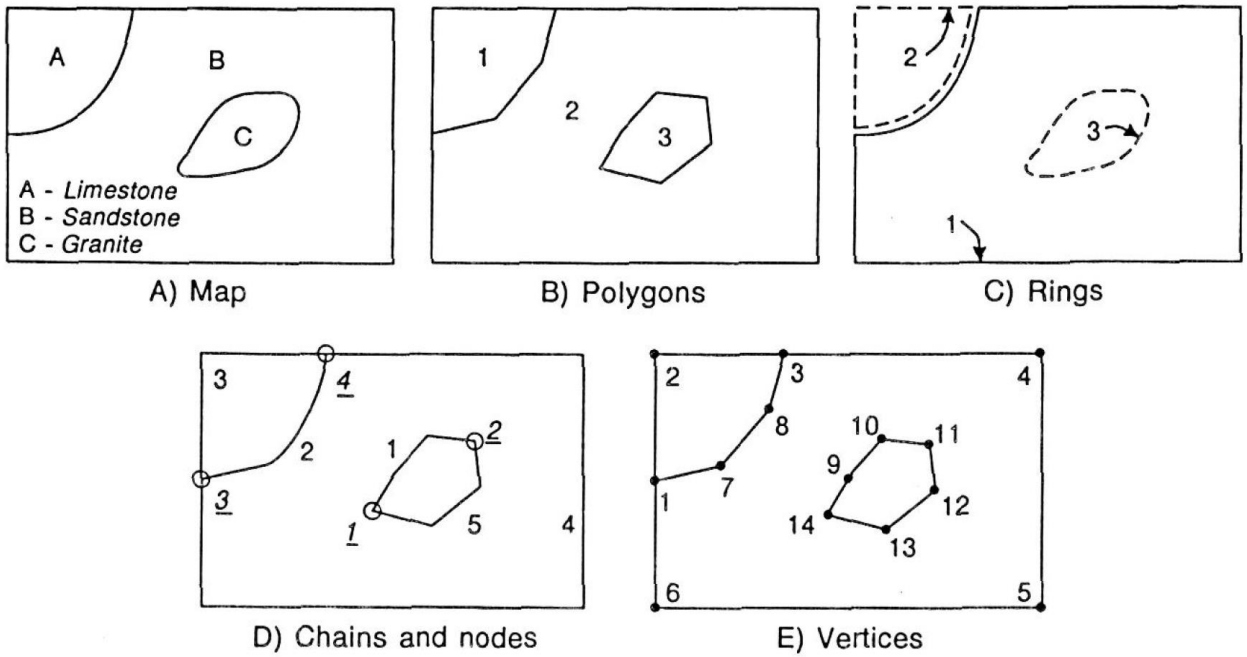
Topologische Datenstrukturen: Darstellung von “Nachbarschafts”beziehungen zwischen Objekten (geometrische Eigenschaften und Distanzen spielen keine grosse Rolle, sondern nur die relative Lage und Beziehung zueinander)

- *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



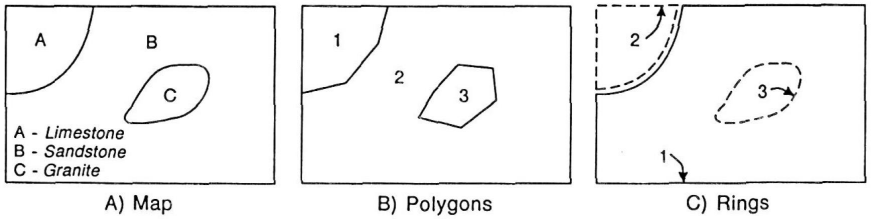
Vektormodell

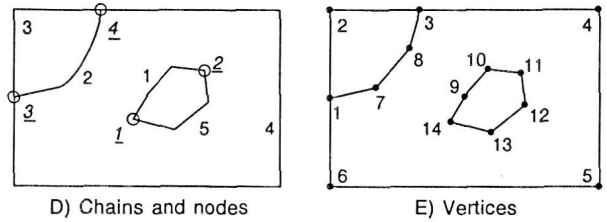
Darstellung der topologischen Struktur mittels Listen/Tabellen (z.B. van-Roessel-Topologie):

- 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

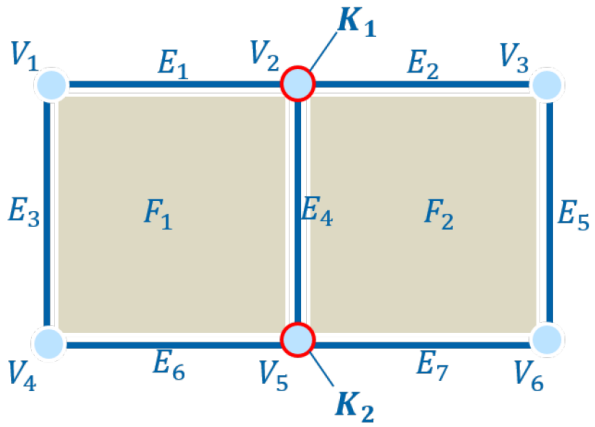
A. Polygon topology table

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



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

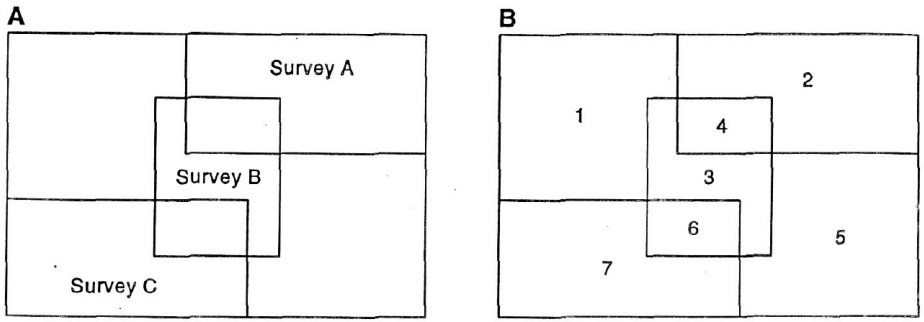
Für welche (beispielhaften) Abfragen könnte solch ein topologische Struktur sinnvoll/hilfreich sein?

- 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

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	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

SURVEY

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

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)

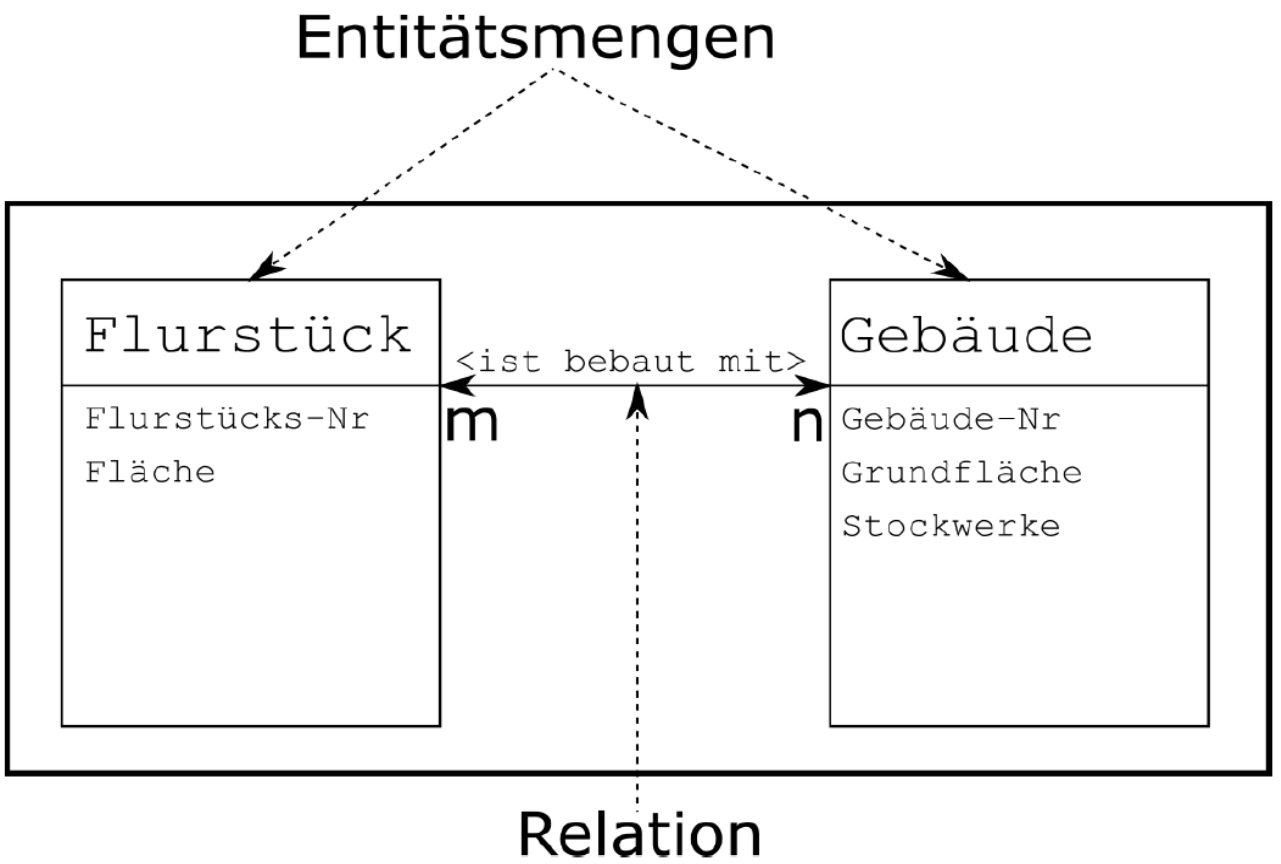
Weitere Datenmodelle

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 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

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:

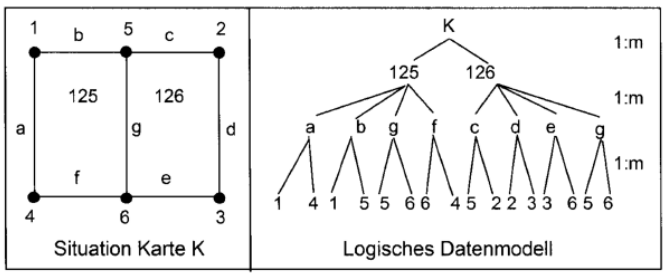


Abbildung 6.43: Hierarchisches Datenmodell.

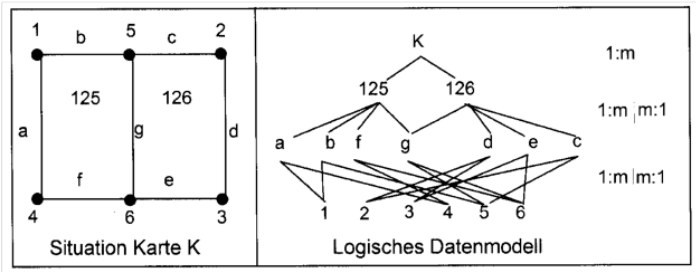
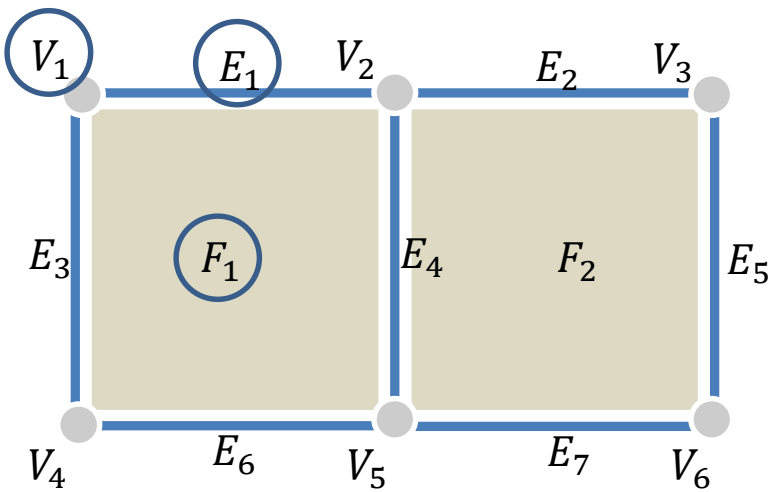
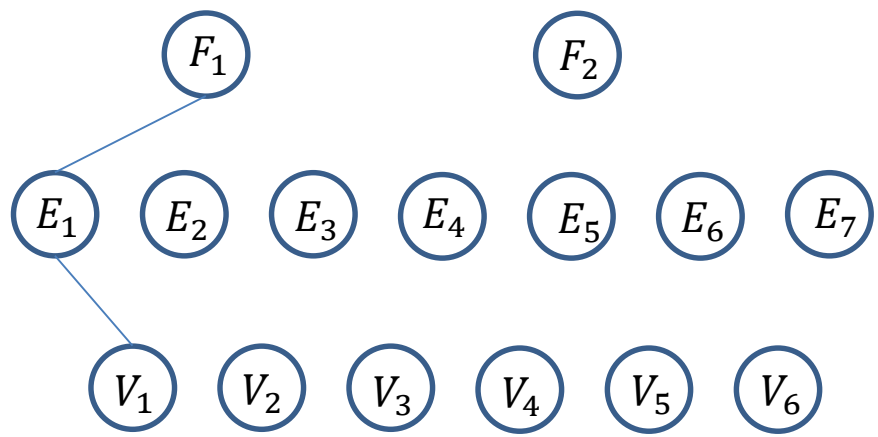


Abbildung 6.44: Netzwerkartiges Datenmodell.

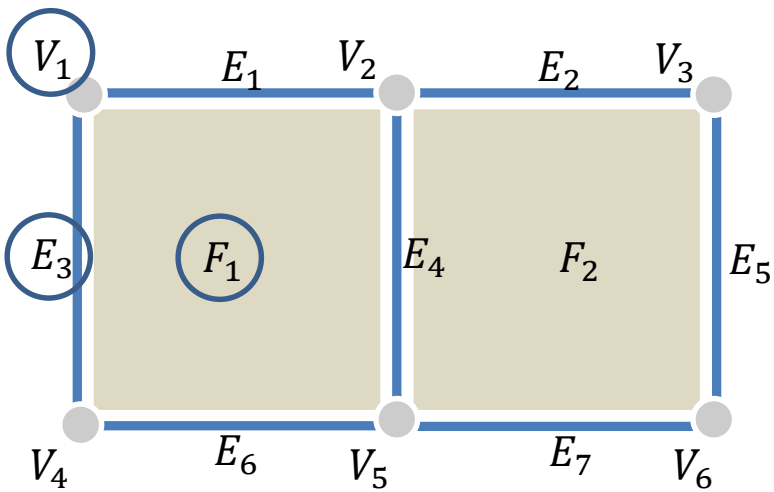
Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell



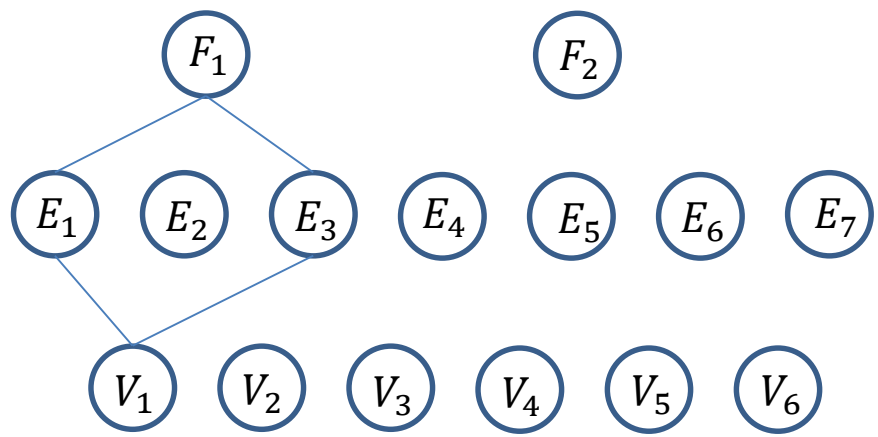
$\{V_1, E_1, F_1\}$



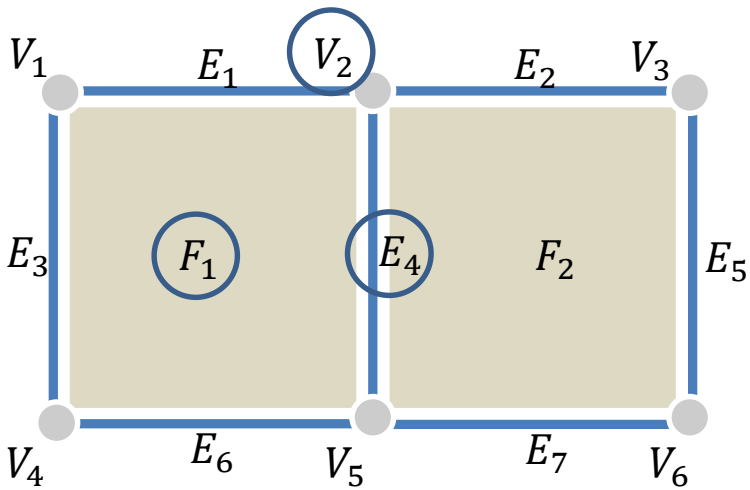
Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell



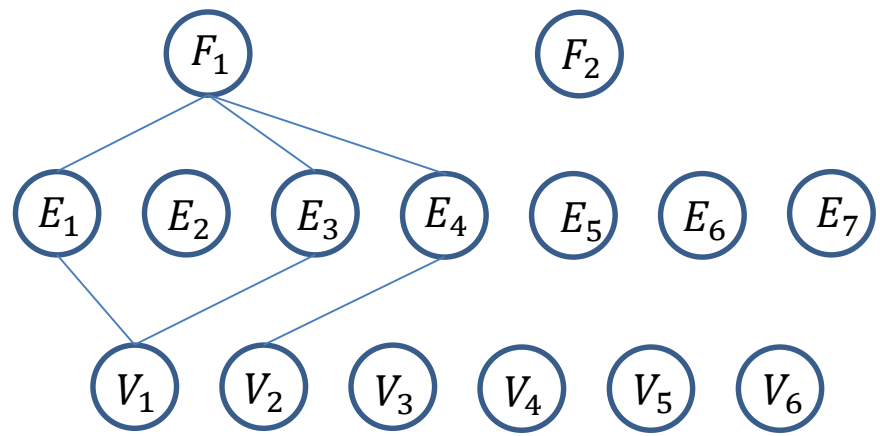
$\{V_1, E_3, F_1\}$



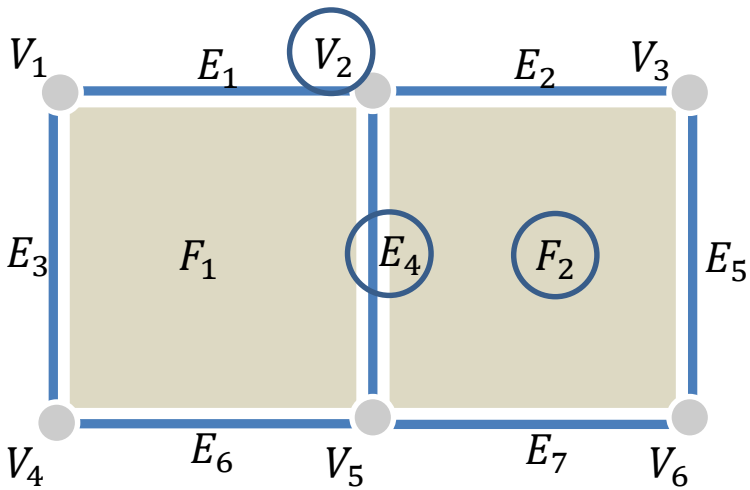
Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell



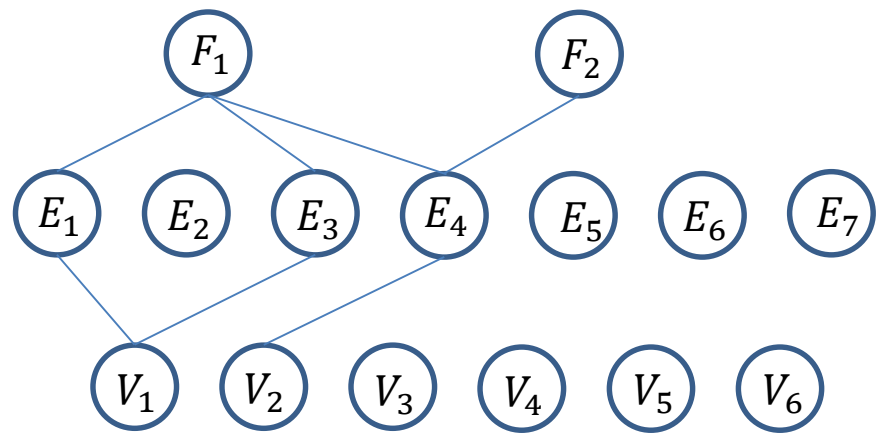
$\{V_2, E_4, F_1\}$



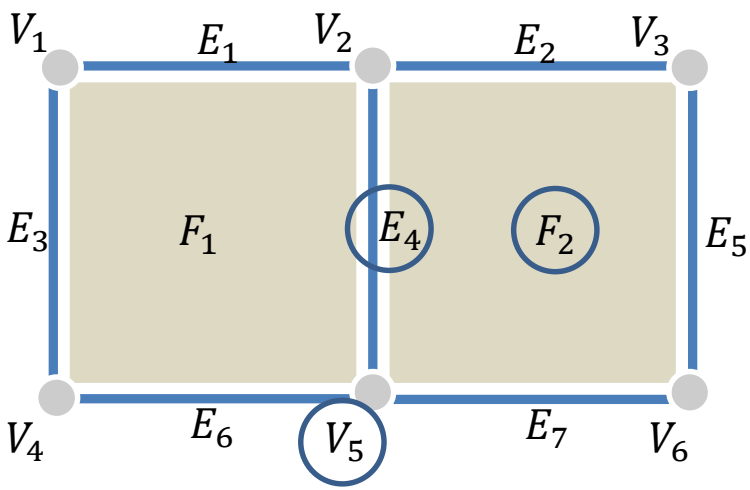
Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell



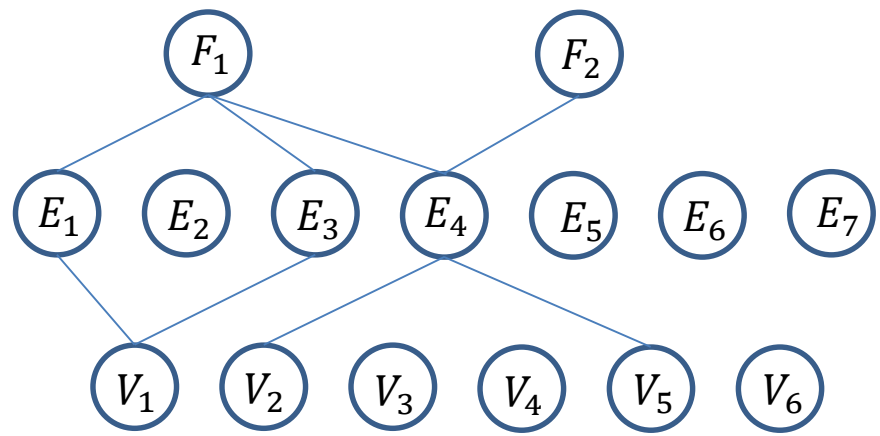
$\{V_2, E_4, F_2\}$



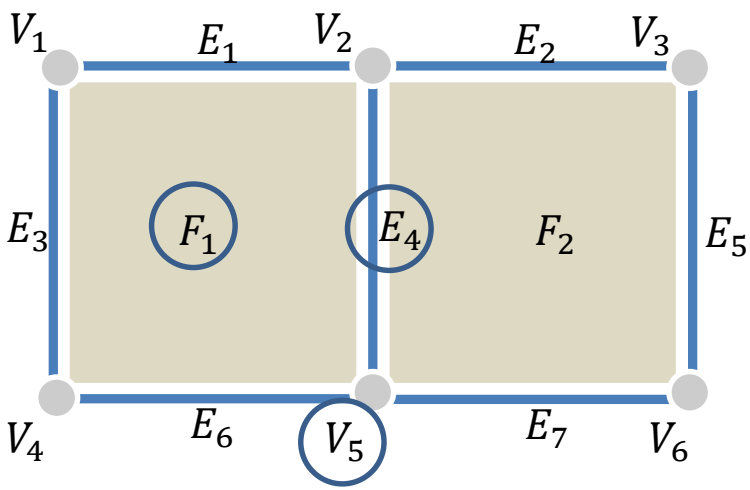
Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell



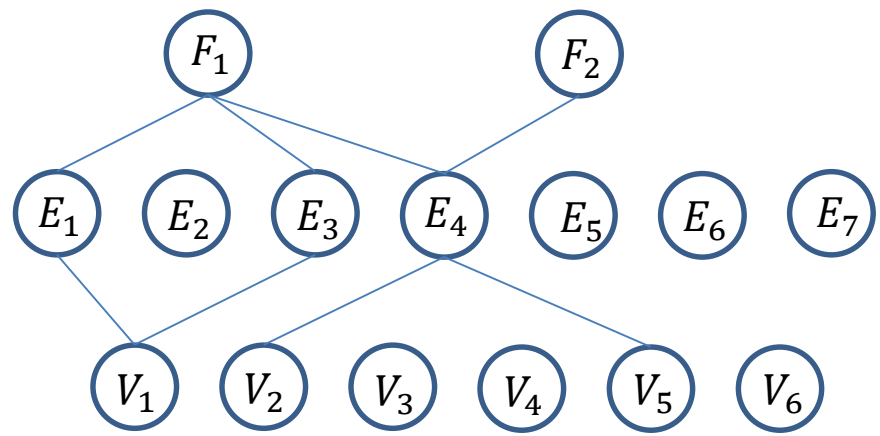
$\{V_5, E_4, F_2\}$



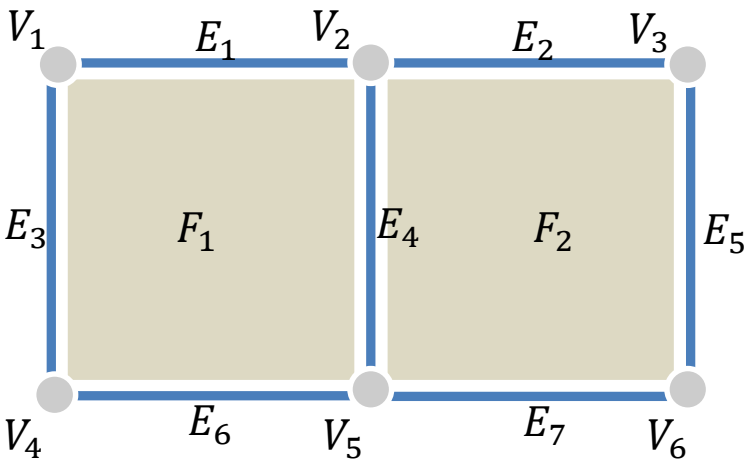
Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell



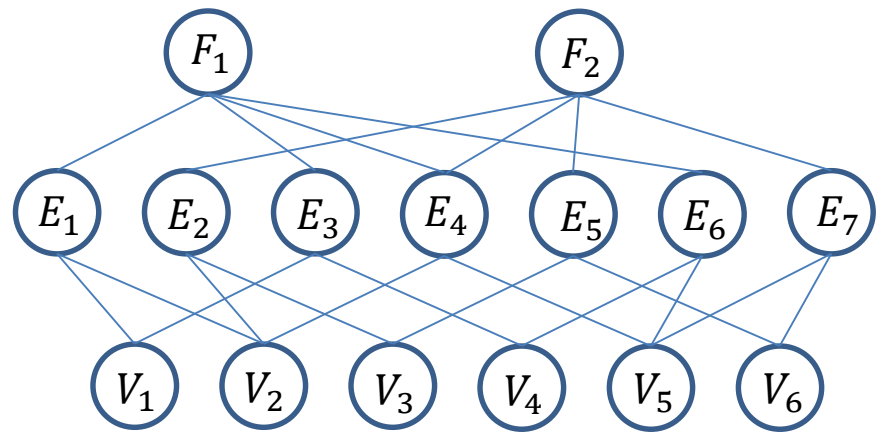
$\{V_5, E_4, F_1\}$



Inzidenzgraph als Repräsentant für ein Graphen-basiertes Datenmodell



Kompletter Inzidenzgraph



⇒ **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

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

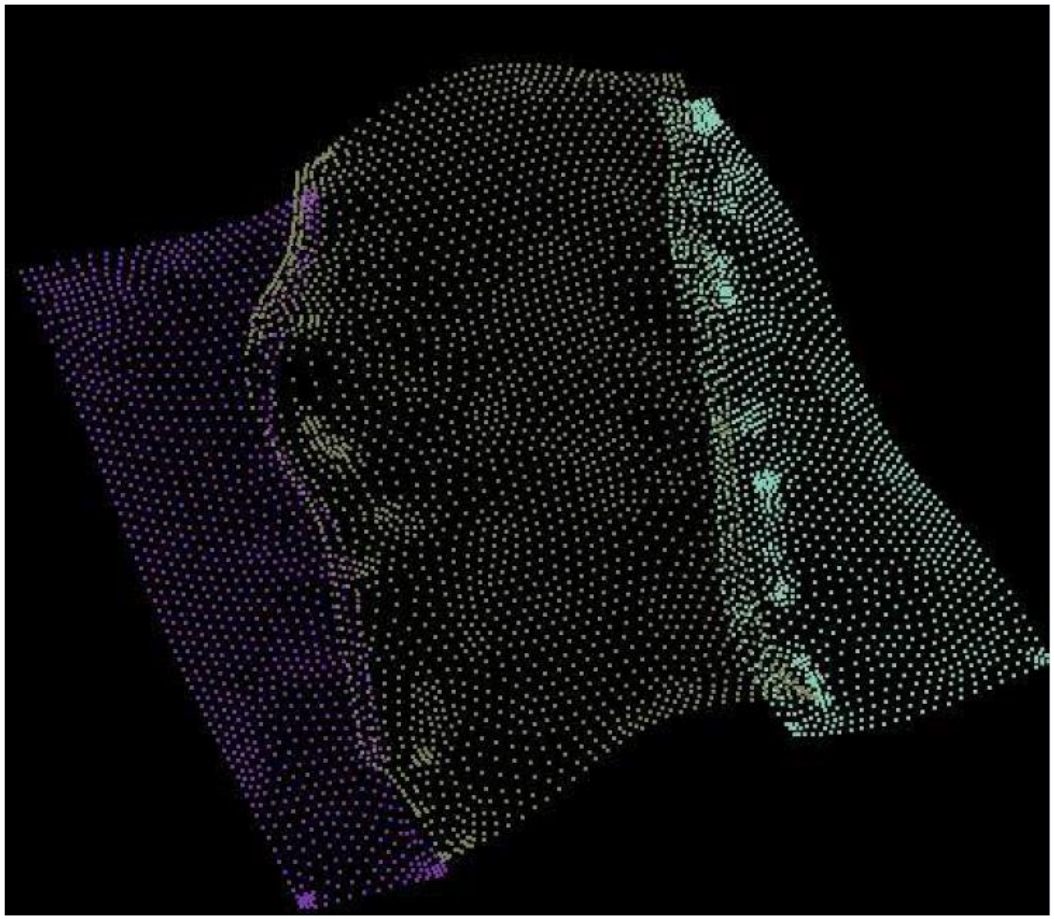
LITHOLOGY relation
(1 insertion shown in italics)

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

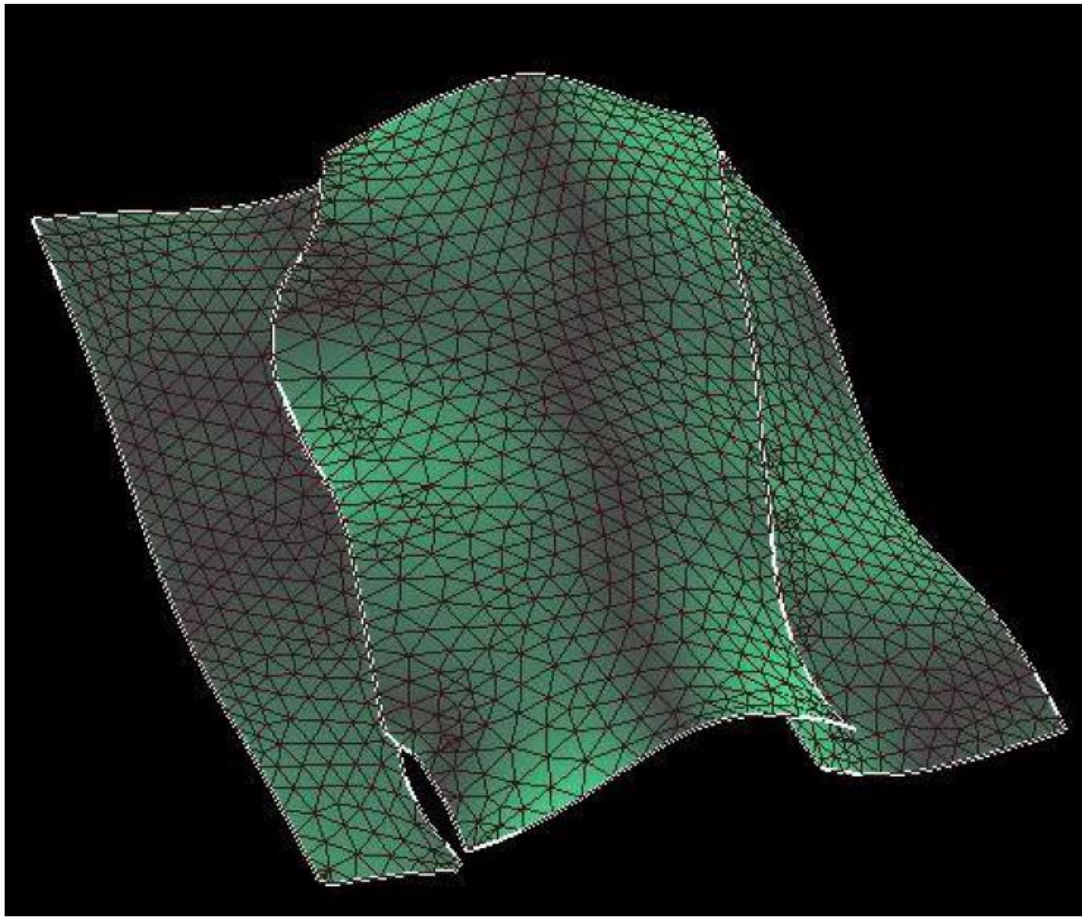
Übersicht

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)

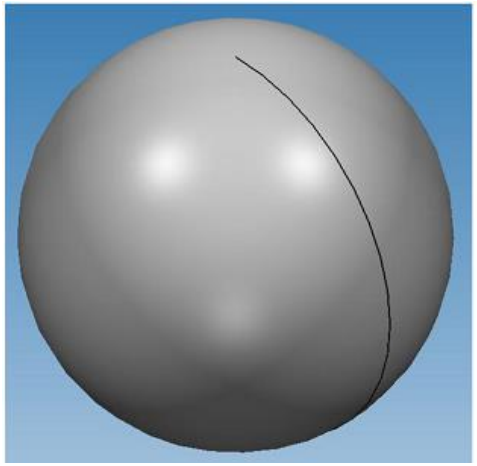


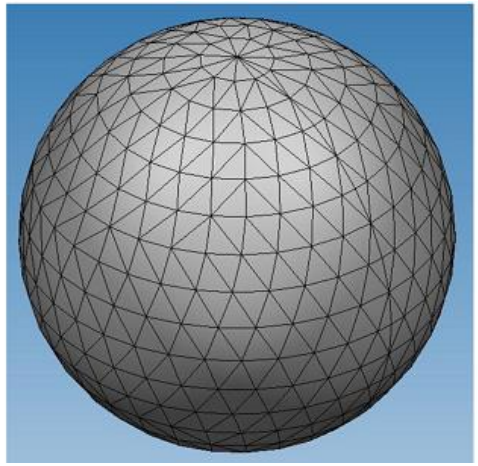
Zerlegungen und Triangulierung (Partition and Triangulation)



Zerlegungen und Triangulierung (Partition and Triangulation)

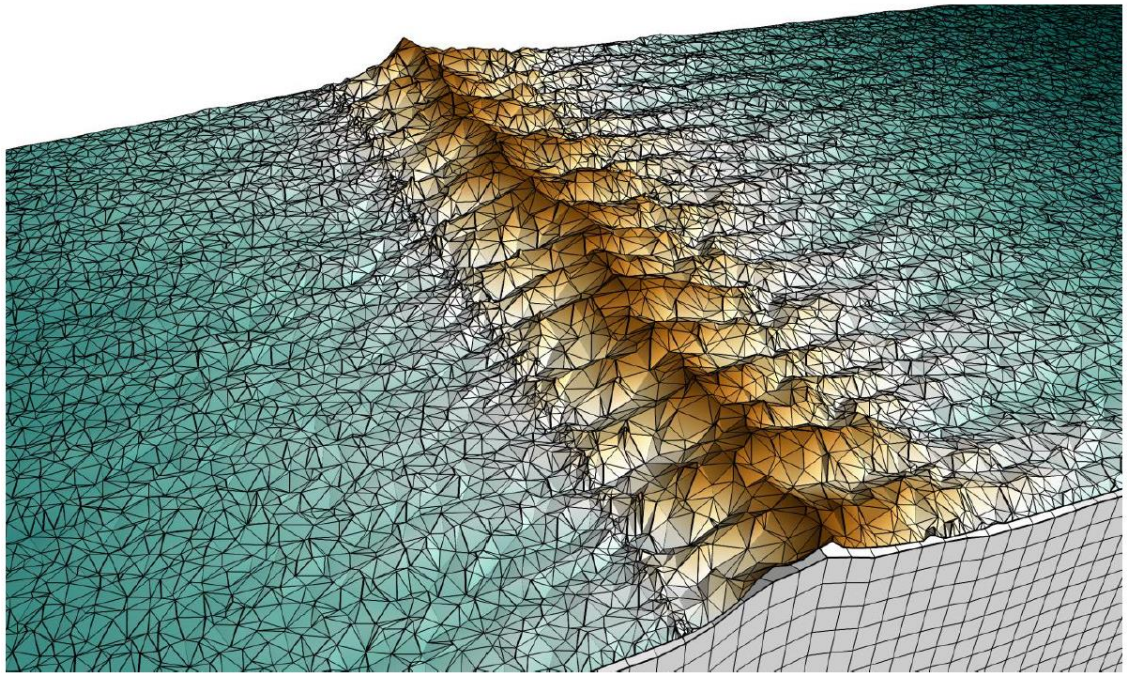
Im Bereich “computer aided geometric design” oder “computational geometry” bezieht sich der Terminus “Zerlegung” auf jedes beliebige Verfahren, um ein beliebiges n-dimensionales Objekt in eine Menge benachbarter polytopaler* n-dimensionaler Zellen zu zerlegen.





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.

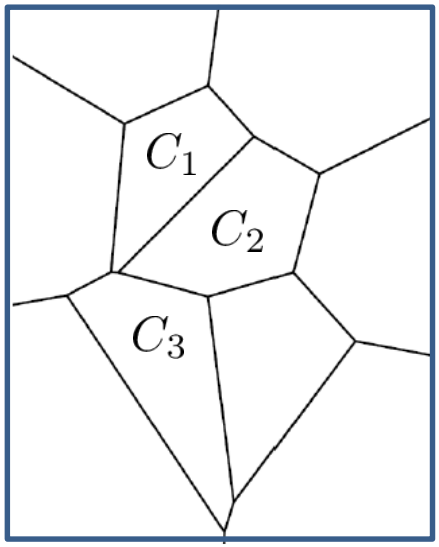
Partition / Tesselation

A **Partition** of a region $\Omega \subset \mathbb{R}^n$ is a set of open cells $C_i \subset \mathbb{R}^n$, $i = 1, \ldots, N$, with $C_i \cap C_j = \emptyset$, $i \neq j$, such that

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

Partitions are also called **Tesselations**.

- Partition → Zerlegung
- Tesselation → Vermaschung



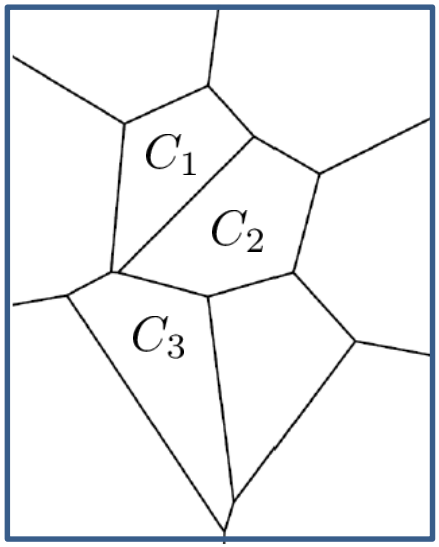
Partition / Tesselation

A **Partition** of a region $\Omega \subset \mathbb{R}^n$ is a set of open cells $C_i \subset \mathbb{R}^n$, $i = 1, \ldots, N$, with $C_i \cap C_j = \emptyset$, $i \neq j$, such that

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

Partitions are also called **Tesselations** (Vermaschungen).

The **openeness** needs to be considered for a precise mathematical formulation $\longrightarrow$ later!

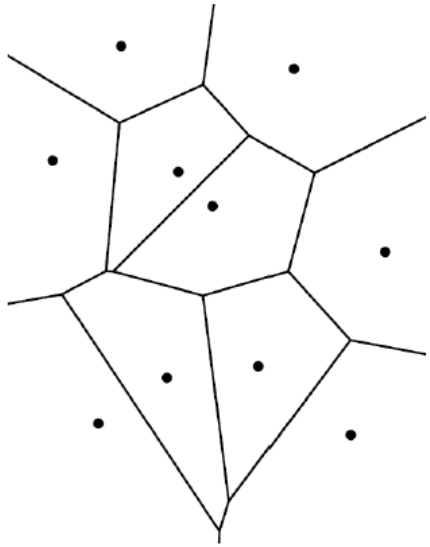


Voronoi Tesselation

Let $\{p_i \in \mathbb{R}^n : i = 1, \ldots, N\}$ be a set of points. A **Voronoi cell** V_i denotes the set

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

Such a cell is a particular form of **Polyhedron** (or polygon, in 2D). Furthermore, the cells $C_i = V_i$, $i = 1, \ldots, N$ "nearly" form a partition of $\mathbb{R}^n$.



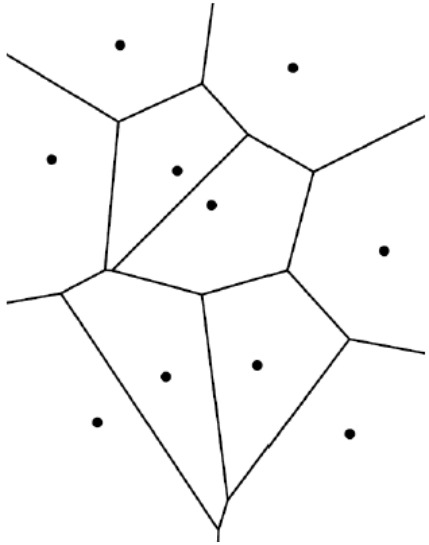
Voronoi Tesselation

Let $\{p_i \in \mathbb{R}^n : i = 1, \ldots, N\}$ be a set of points. A **Voronoi cell** V_i denotes the set

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

Such a cell is a particular form of **Polyhedron** (or polygon, in 2D). Furthermore, the cells $C_i = V_i$, $i = 1, \ldots, N$ "nearly" form a partition of $\mathbb{R}^n$.

Why just "nearly"? Formally, also the edges of the Voronoi cells need to be included, yielding further cells C_i, $i = N + 1, \ldots, N + M$, as well as the points at the intersections of edges, yielding yet further cells C_i, $i = N + M + 1, \ldots, N + M + K$. This is a result of the openness condition.



Voronoi Tesselation

Let $\{p_i \in \mathbb{R}^n : i = 1, \ldots, N\}$ be a set of points. A **Voronoi cell** V_i denotes the set

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

Such a cell is a particular form of **Polyhedron** (or polygon, in 2D). Furthermore, the cells $C_i = V_i$, $i = 1, \ldots, N$ "nearly" form a partition of $\mathbb{R}^n$.

The **bisector** of two points p_i, p_j is denoted by

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

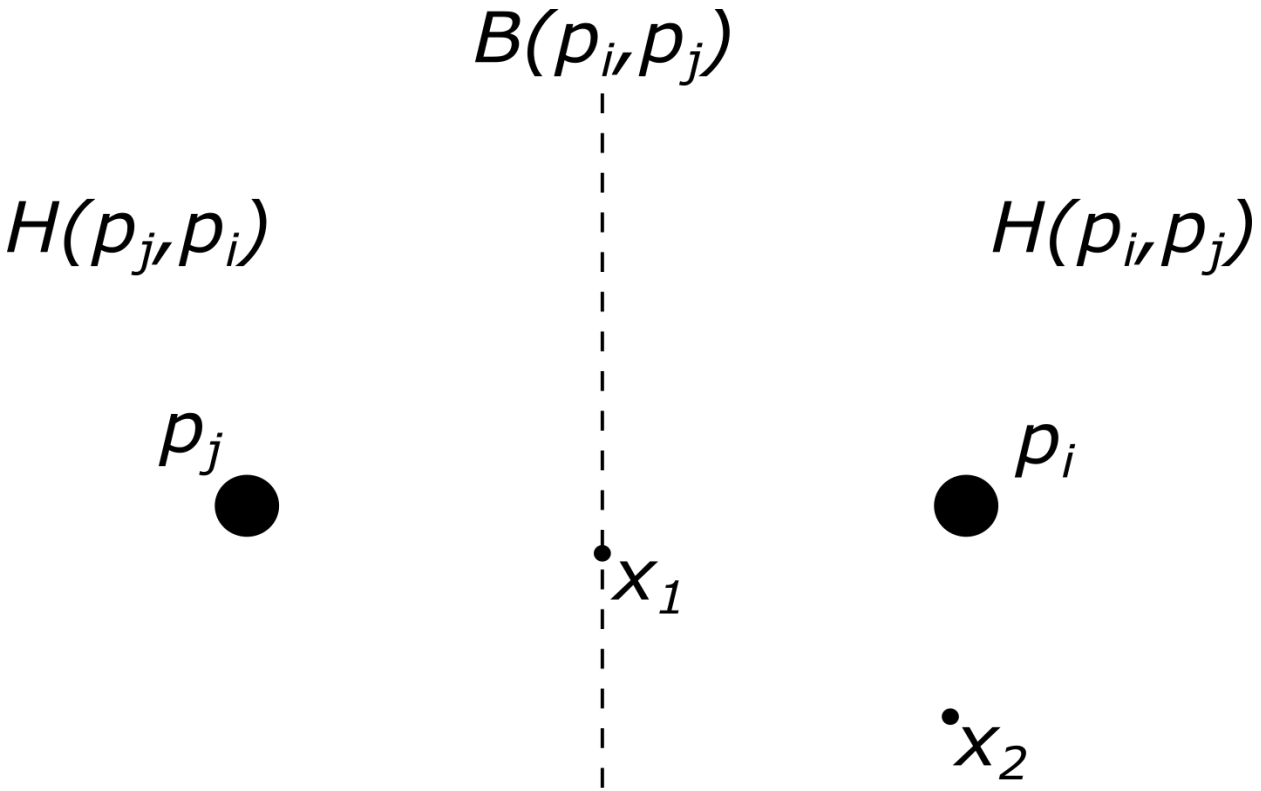
and separates $\mathbb{R}^n$ into two half-spacee. In particular, it represents the boundary of the half-space

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

Each Voronoi cell can then be expressed in the form

$$V_i = \bigcap_{\substack{j=1,\ldots,N,\\ i \neq j}} H_{i,j}.$$

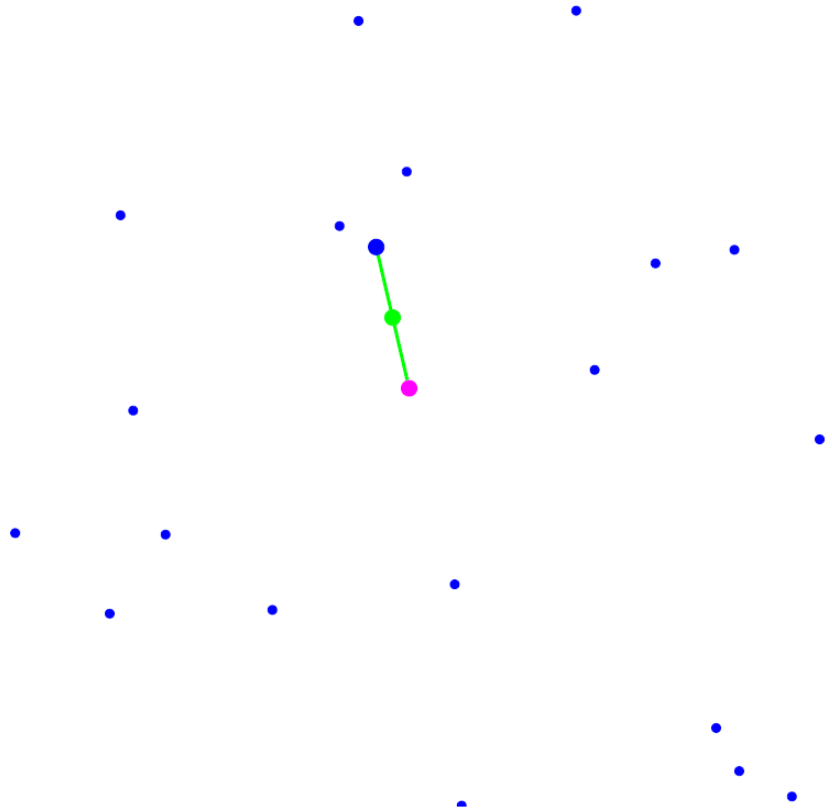
Voronoi Tesselation



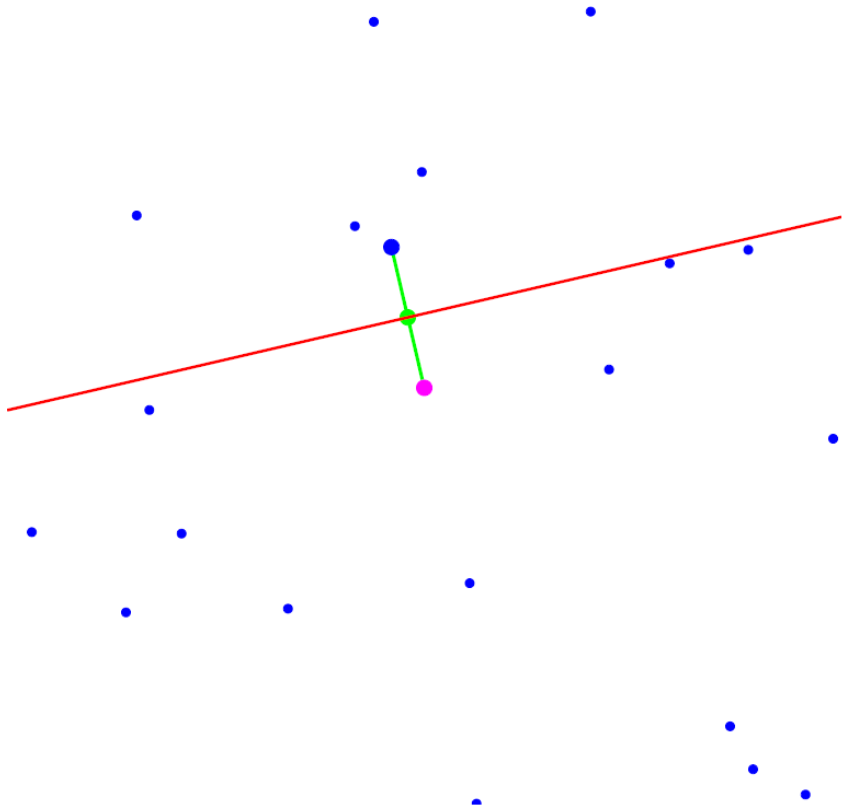
$$|x_1 - p_i| = |x_1 - p_j| \rightarrow x_1 \in B(p_i, p_j)$$

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

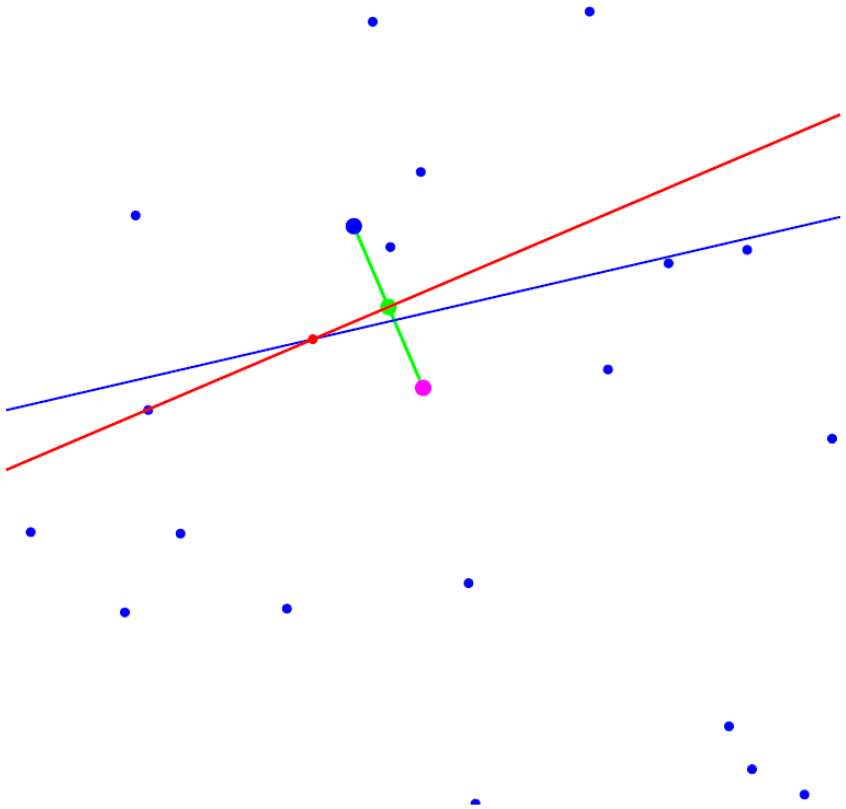
Example: Voronoi Tesselation in 2-D



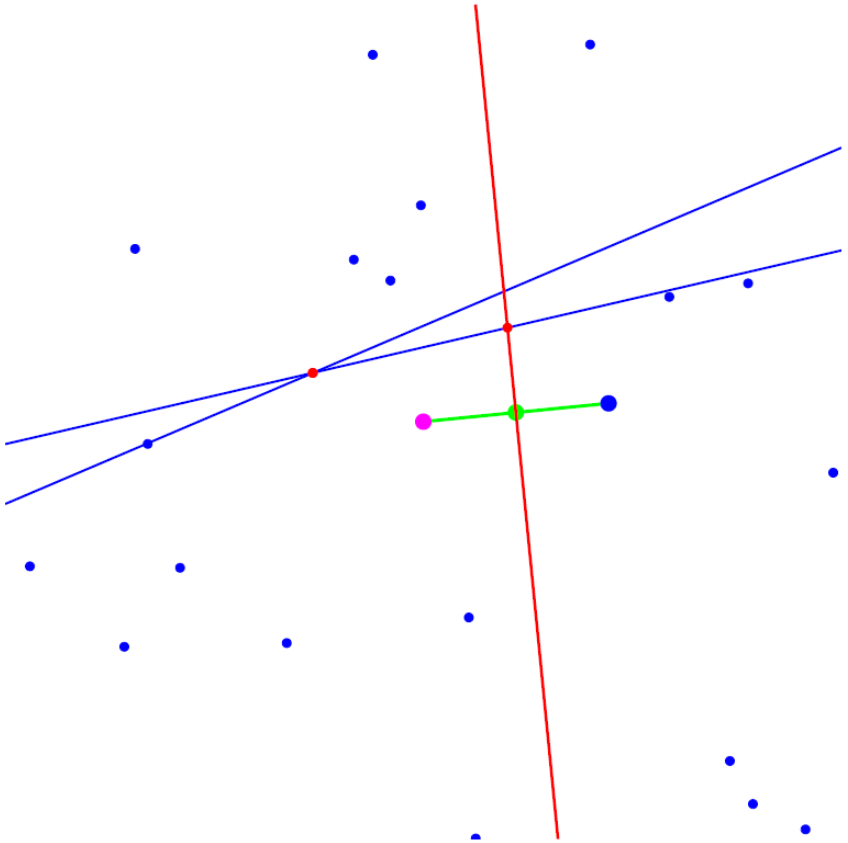
Example: Voronoi Tesselation in 2-D



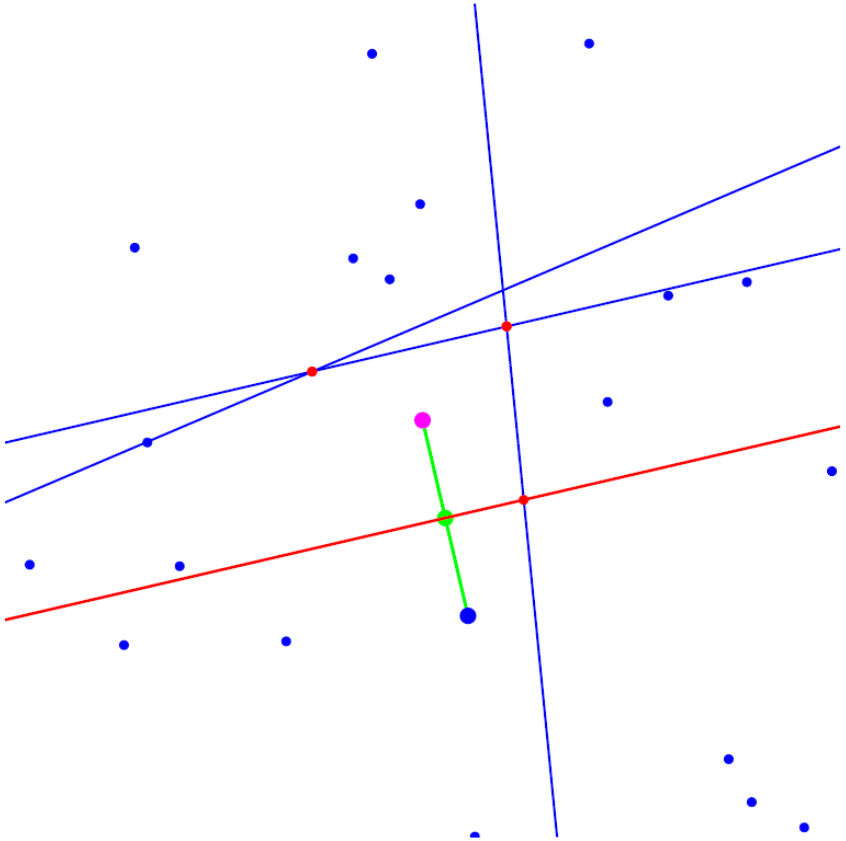
Example: Voronoi Tesselation in 2-D



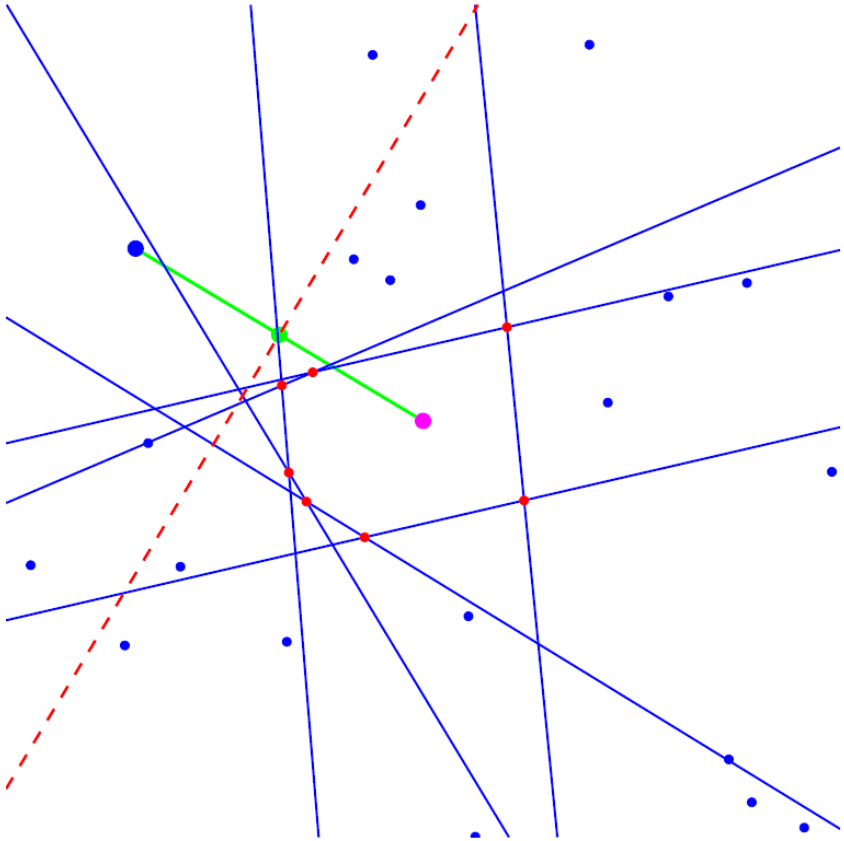
Example: Voronoi Tesselation in 2-D



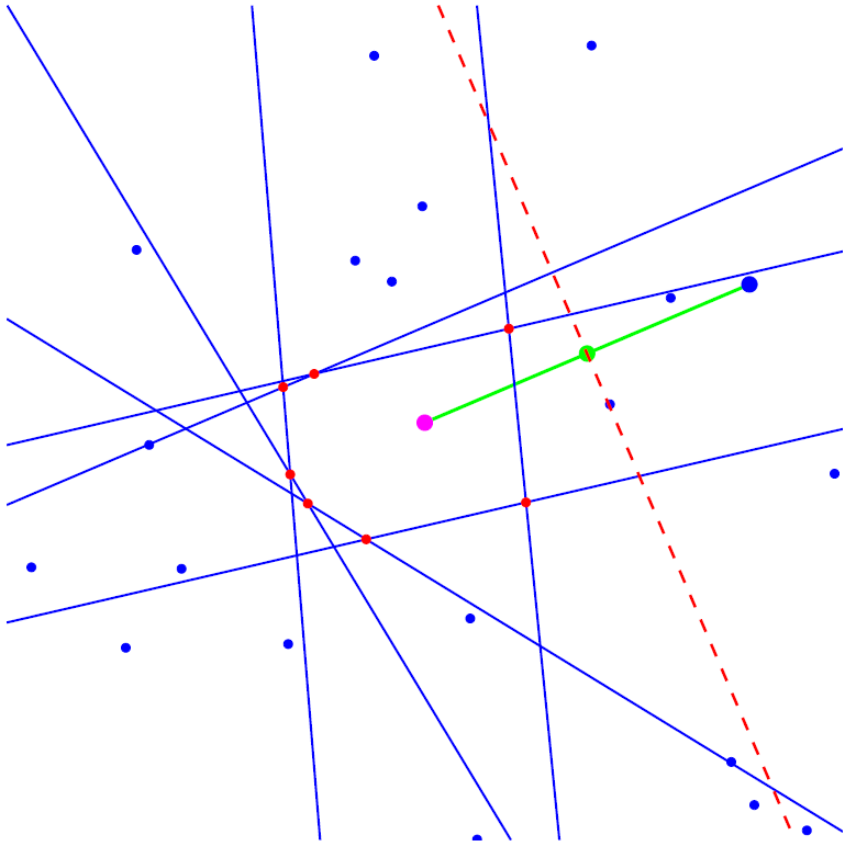
Example: Voronoi Tesselation in 2-D



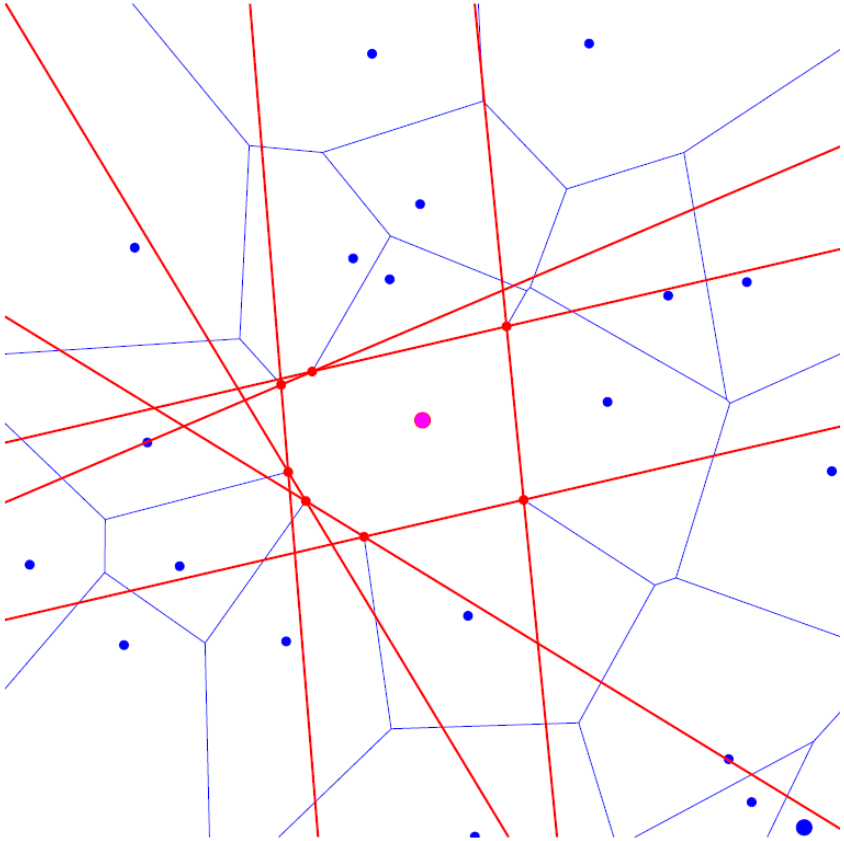
Example: Voronoi Tesselation in 2-D



Example: Voronoi Tesselation in 2-D



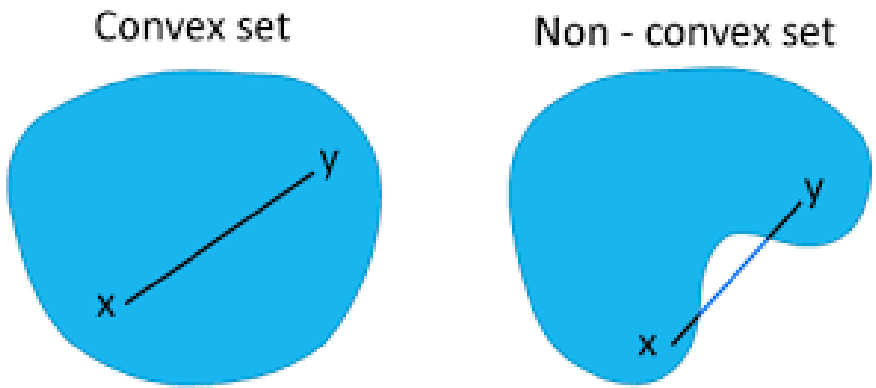
Example: Voronoi Tesselation in 2-D



Voronoi Tesselation

Lemma (Properties). Let $\mathcal{V} = \{V_1, \ldots, V_N\}$ denote the Voronoi Tesselation of the point set $P = \{p_1, \ldots, p_N\} \subset \mathbb{R}^d$ in general position. Then the following properties hold true:

(a) Every cell V_i is **convex** (i.e., if $x, y \in V_i$, then $\lambda x + \mu x \in V_i$ for all $\lambda, \mu \in [0, 1]$ with $\lambda + \mu \leq 1$).

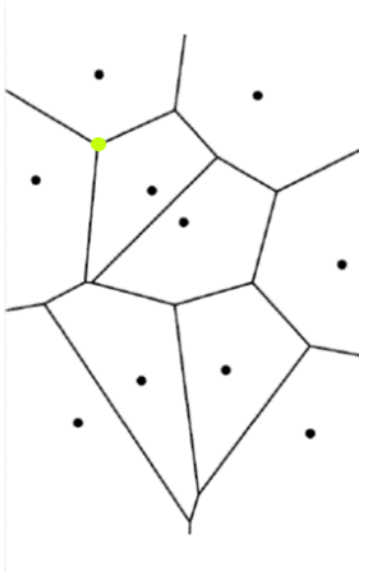


Voronoi Tesselation

Lemma (Properties). Let $\mathcal{V} = \{V_1, \ldots, V_N\}$ denote the Voronoi Tesselation of the point set $P = \{p_1, \ldots, p_N\} \subset \mathbb{R}^d$ in general position. Then the following properties hold true:

(a) Every cell V_i is **convex** (i.e., if $x, y \in V_i$, then $\lambda x + \mu x \in V_i$ for all $\lambda, \mu \in [0, 1]$ with $\lambda + \mu \leq 1$).

(b) Every vertex of a Voronoi cell V_i lies on the boundary of exactly $d+1$ different cells V_j.



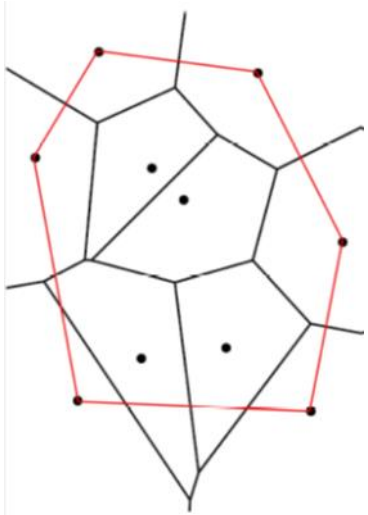
Voronoi Tesselation

Lemma (Properties). Let $\mathcal{V} = \{V_1, \ldots, V_N\}$ denote the Voronoi Tesselation of the point set $P = \{p_1, \ldots, p_N\} \subset \mathbb{R}^d$ in general position. Then the following properties hold true:

(a) Every cell V_i is **convex** (i.e., if $x, y \in V_i$, then $\lambda x + \mu x \in V_i$ for all $\lambda, \mu \in [0, 1]$ with $\lambda + \mu \leq 1$).

(b) Every vertex of a Voronoi cell V_i lies on the boundary of exactly $d + 1$ different cells V_j.

(c) The cell V_i is unbounded if and only if p_i lies on the boundary of the convex hull of P (the convex hull of P is given by $\mathcal{C}(P) = \left\{ \sum_{j=1}^{N} \lambda_j p_j : \lambda_j \in [0, 1], \sum_{j=1}^{N} \lambda_j \leq 1 \right\}$).



Voronoi Tesselation

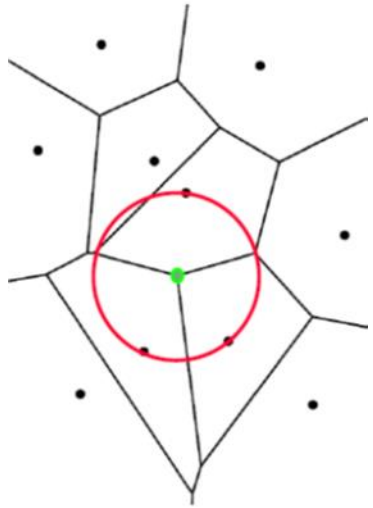
Lemma (Properties). Let $\mathcal{V} = \{V_1, \ldots, V_N\}$ denote the Voronoi Tesselation of the point set $P = \{p_1, \ldots, p_N\} \subset \mathbb{R}^d$ in general position. Then the following properties hold true:

(a) Every cell V_i is **convex** (i.e., if $x, y \in V_i$, then $\lambda x + \mu x \in V_i$ for all $\lambda, \mu \in [0, 1]$ with $\lambda + \mu \leq 1$).

(b) Every vertex of a Voronoi cell V_i lies on the boundary of exactly $d + 1$ different cells V_j.

(c) The cell V_i is unbounded if and only if p_i lies on the boundary of the convex hull of P (the convex hull of P is given by $\mathcal{C}(P) = \left\{ \sum_{j=1}^{N} \lambda_j p_j : \lambda_j \in [0, 1], \sum_{j=1}^{N} \lambda_j \leq 1 \right\}$).

(d) If p is the vertex belonging to the cells $V_{i_1}, \ldots, V_{i_{d+1}}$, then p is the center of the sphere that passes through the points $p_{i_1}, \ldots, p_{i_{d+1}}$. There lie no further points of P within this sphere.



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

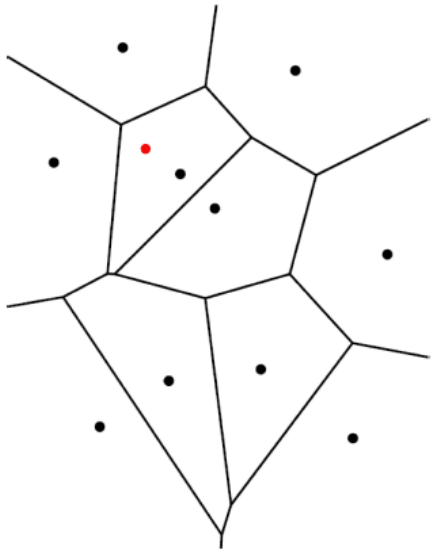
cellCenters		
cellID	**x_center**	**y_center**
1	x_1	y_1
⋮	⋮	⋮
max. Anzahl der zellen m	x_m	y_m

Interpolation mittels Voronoi-Zerlegung: Nearest Neighbor Interpolation

Let $P = \{p_i \in \mathbb{R}^n : i = 1, \ldots, N\}$ a set of points in general position (i.e., there are no $n+1$ points of P that lie on an $n-1$-sphere).

A measurement/datum h_i is assigned to every point p_i.

Nearest Neighbor Interpolation: For every $p \in \mathbb{R}^n$ there exists one and only one Voronoi cell V_i with $p \in V_i$. Then, the point p is assigned the value $h(p) = h_i$.



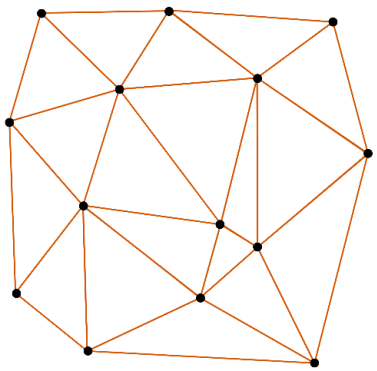
Graphs

A **graph** $\mathcal{G} = (V, E)$ is a pair of abstracte vertices $V = \{v_1, \ldots, v_N\}$ and edges $E = \{e_1, \ldots, e_M\}$, which each connect two vertices. A graph is called "simple" if

1.) there is at most one edge connecting two vertices,

2.) edges have no orientation,

3.) there exist no loops (i.e., there exists no edge with identical starting and endpoint).



Graphs

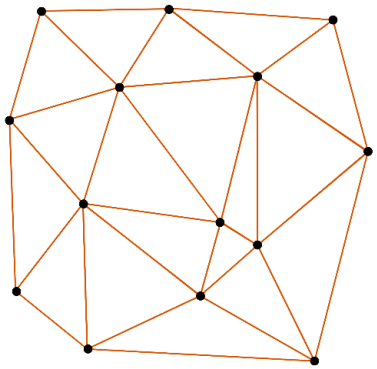
A **graph** $\mathcal{G} = (V, E)$ is a pair of abstracte vertices $V = \{v_1, \ldots, v_N\}$ and edges $E = \{e_1, \ldots, e_M\}$, which each connect two vertices. A graph is called "simple" if

1.) there is at most one edge connecting two vertices,

2.) edges have no orientation,

3.) there exist no loops (i.e., there exists no edge with identical starting and endpoint).

Vertices are **0-cells**, edges **1-cells**, and enclosed faces **2-cells**. Edges determine the "neighborhood" of faces, and vertices determine the "neighborhood" of edges.



Graphs

A graph can be described by its **adjacency matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, \text{ if } i \neq j \text{ and } v_i, v_j \text{ are connected by an edge} \\ 0, \text{ else} \end{cases}$$

Graphs

A graph can be described by its **adjacency matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, \text{ if } i \neq j \text{ and } v_i, v_j \text{ are connected by an edge} \\ 0, \text{ else} \end{cases}$$

An equivalent characterization is given by its **incidence matrix** $I_V \in \mathbb{R}^{N \times M}$:

$$(I_{VE})_{i,j} = \begin{cases} 1, \text{ if } v_i \text{ is and endpoint of the edge } e_j \\ 0, \text{ else} \end{cases}$$

Each column of the incidence matrix of a simple graph contains the value 1 exactly twice.

Graphs

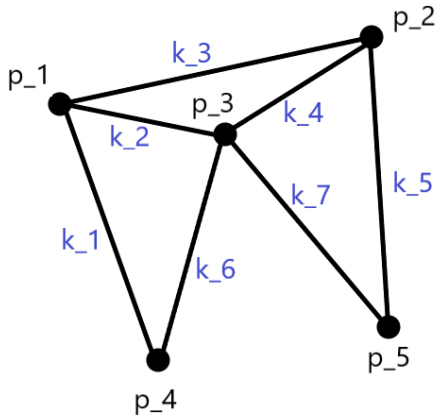
A graph can be described by its **adjacency matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, \text{ if } i \neq j \text{ and } v_i,\, v_j \text{ are connected by an edge} \\ 0, \text{ else} \end{cases}$$

An equivalent characterization is given by its **incidence matrix** $I_V \in \mathbb{R}^{N \times M}$:

$$(I_{VE})_{i,j} = \begin{cases} 1, \text{ if } v_i \text{ is and endpoint of the edge } e_j \\ 0, \text{ else} \end{cases}$$

Each column of the incidence matrix of a simple graph contains the value 1 exactly twice.



Adjacency matrix of the exemplary graph:

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

Graphs

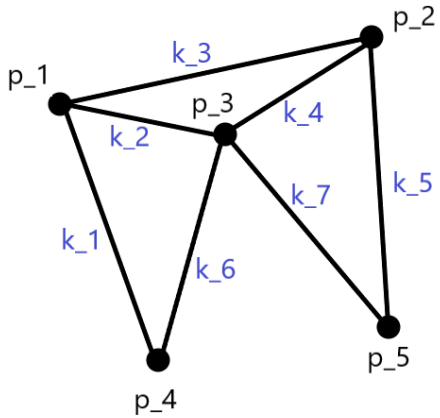
A graph can be described by its **adjacency matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, \text{ if } i \neq j \text{ and } v_i,\, v_j \text{ are connected by an edge} \\ 0, \text{ else} \end{cases}$$

An equivalent characterization is given by its **incidence matrix** $I_V \in \mathbb{R}^{N \times M}$:

$$(I_{VE})_{i,j} = \begin{cases} 1, \text{ if } v_i \text{ is and endpoint of the edge } e_j \\ 0, \text{ else} \end{cases}$$

Each column of the incidence matrix of a simple graph contains the value 1 exactly twice.



Incidence matrix of the exemplary graph:

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

Graphs

A graph can be described by its **adjacency matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, \text{ if } i \neq j \text{ and } v_i, v_j \text{ are connected by an edge} \\ 0, \text{ else} \end{cases}$$

An equivalent characterization is given by its **incidence matrix** $I_V \in \mathbb{R}^{N \times M}$:

$$(I_{VE})_{i,j} = \begin{cases} 1, \text{ if } v_i \text{ is and endpoint of the edge } e_j \\ 0, \text{ else} \end{cases}$$

Each column of the incidence matrix of a simple graph contains the value 1 exactly twice.

It holds

$$(A_V)_{i,j} = \begin{cases} (I_{VE} I_{VE}^T)_{i,j}, & \text{if } i \neq j, \\ 0, & \text{else.} \end{cases}$$

Topology of a Voronoi Tesselation

Aside of edges and vertices, a (2-D) Voronoi tesselation contains **faces** $F_1, \ldots, F_K$. The corresponding incidence matrices can be represented as follows:

$$(I_{VE})_{i,j} = \begin{cases} 1, \text{ if } v_i \text{ is an endpoint of the edge } e_j \\ 0, \text{ else} \end{cases}$$

$$(I_{EF})_{i,j} = \begin{cases} 1, \text{ if } e_i \text{ is an edge of the boundary of } F_j \\ 0, \text{ else} \end{cases}$$

$$(I_{VF})_{i,j} = (I_{VE} I_{EF})_{i,j} \begin{cases} 1, \text{ if } v_i \text{ is a vertex on the boundary of } F_j \\ 0, \text{ else} \end{cases}$$

Topology of a Voronoi Tesselation

Aside of edges and vertices, a (2-D) Voronoi tesselation contains **faces** $F_1, \ldots, F_K$. The corresponding incidence matrices can be represented as follows:

$$(I_{VE})_{i,j} = \begin{cases} 1, \text{ if } v_i \text{ is an endpoint of the edge } e_j \\ 0, \text{ else} \end{cases}$$

$$(I_{EF})_{i,j} = \begin{cases} 1, \text{ if } e_i \text{ is an edge of the boundary of } F_j \\ 0, \text{ else} \end{cases}$$

$$(I_{VF})_{i,j} = (I_{VE} I_{EF})_{i,j} \begin{cases} 1, \text{ if } v_i \text{ is a vertex on the boundary of } F_j \\ 0, \text{ else} \end{cases}$$

These matrices determine uniquely which vertices, edges, and faces are neighbours. In particular, the adjacency of two Voronoi cells is described by the matrix:

$$(A_F)_{i,j} = \begin{cases} (I_{EF}^T I_{EF})_{i,j}, & \text{if } i \neq j \\ 0, & \text{else.} \end{cases}$$

Dual Graph

The **dual graph** $\mathcal{G}^* = (V^*, E^*)$ with respect to $\mathcal{G}$ is defined by identifying the vertices V^* with the faces F. The set of edges E^* is defined by including an edge between two vertices of V^* if the faces of the original graph were adjacent. In particular, the adjacency matrix A_{V^*} is identical to A_F.

Dual Graph

The **dual graph** $\mathcal{G}^* = (V^*, E^*)$ with respect to $\mathcal{G}$ is defined by identifying the vertices V^* with the faces F. The set of edges E^* is defined by including an edge between two vertices of V^* if the faces of the original graph were adjacent. In particular, the adjacency matrix A_{V^*} is identical to A_F.

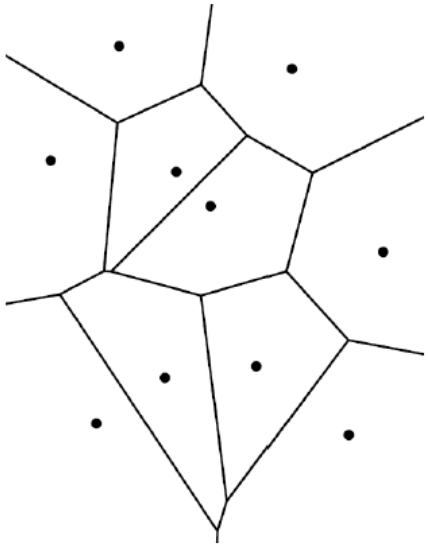
More general: the dual of a n-dimensional cellular partition is defined by identifying the i-cells with the $(n-i)$-cells of the original partition.

Dual Graph

The **dual graph** $\mathcal{G}^* = (V^*, E^*)$ with respect to $\mathcal{G}$ is defined by identifying the vertices V^* with the faces F. The set of edges E^* is defined by including an edge between two vertices of V^* if the faces of the original graph were adjacent. In particular, the adjacency matrix A_{V^*} is identical to A_F.

More general: the dual of a n-dimensional cellular partition is defined by identifying the i-cells with the $(n-i)$-cells of the original partition.

The dual of the Voronoi tesselation yields the **Delaunay triangulation**.

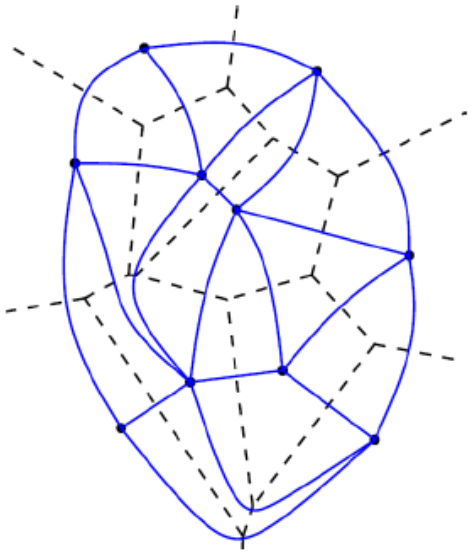


Dual Graph

The **dual graph** $\mathcal{G}^* = (V^*, E^*)$ with respect to $\mathcal{G}$ is defined by identifying the vertices V^* with the faces F. The set of edges E^* is defined by including an edge between two vertices of V^* if the faces of the original graph were adjacent. In particular, the adjacency matrix A_{V^*} is identical to A_F.

More general: the dual of a n-dimensional cellular partition is defined by identifying the i-cells with the $(n-i)$-cells of the original partition.

The dual of the Voronoi tesselation yields the **Delaunay triangulation**.

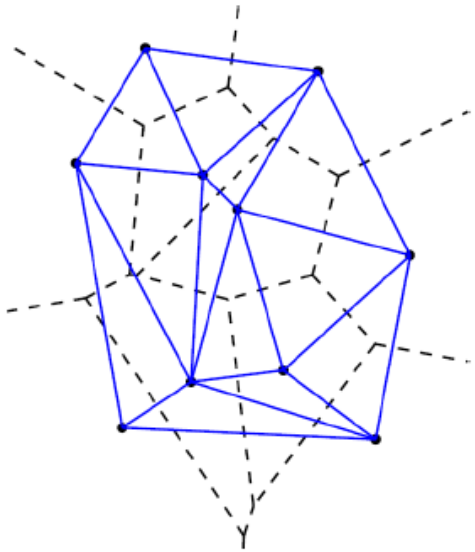


Dual Graph

The **dual graph** $\mathcal{G}^* = (V^*, E^*)$ with respect to $\mathcal{G}$ is defined by identifying the vertices V^* with the faces F. The set of edges E^* is defined by including an edge between two vertices of V^* if the faces of the original graph were adjacent. In particular, the adjacency matrix A_{V^*} is identical to A_F.

More general: the dual of a n-dimensional cellular partition is defined by identifying the i-cells with the $(n-i)$-cells of the original partition.

The dual of the Voronoi tesselation yields the **Delaunay triangulation**.

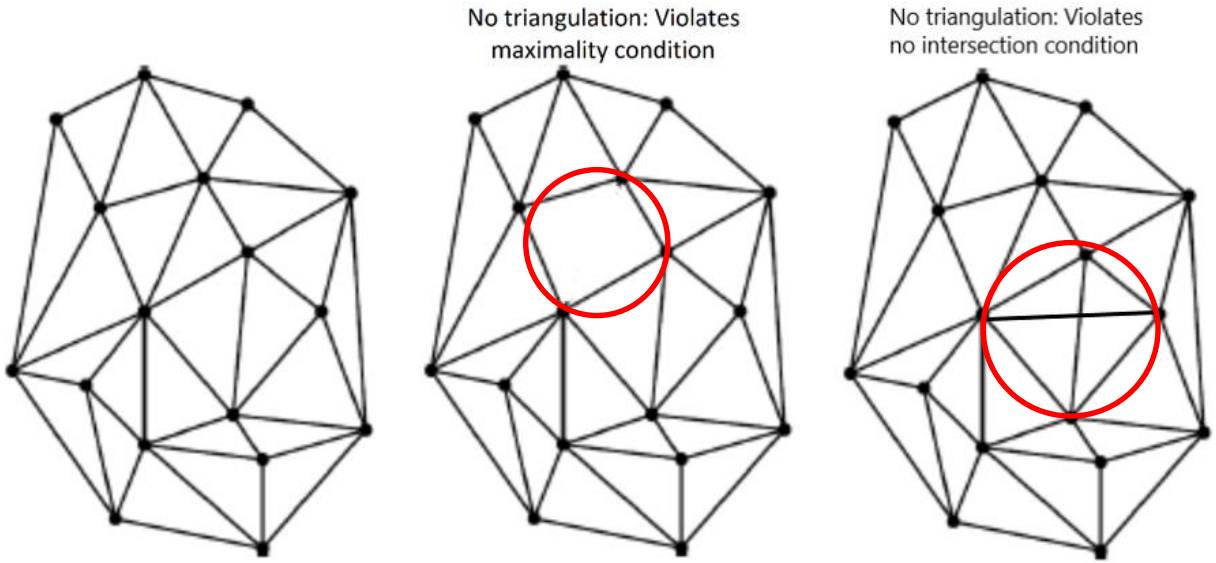


Triangulation

Definition (Triangulation). A triangulation $\mathcal{T}$ of the point cloud $P = \{p_i : i = 1, \ldots, N\} \subset \mathbb{R}^2$ is a maximal family of triangles of the form

$$\mathcal{T} = \{T_i = \mathcal{C}(p_{i_1}, p_{i_2}, p_{i_3}) : \text{the edges } \overline{p_{i_1}p_{i_2}}, \overline{p_{i_1}p_{i_3}}, \overline{p_{i_2}p_{i_3}} \text{ do not intersect with any edge } \overline{p_{k_1}p_{k_2}}, \overline{p_{k_1}p_{k_3}}, \overline{p_{k_2}p_{k_3}} \text{ of any other triangle } T_k,\ k \neq i,\ i, k = 1, \ldots, M_T\}$$

Maximality is meant in the sense that no triangle could be added to $\mathcal{T}$ without violating the 'no intersection'-property.



Triangulation

Definition (Triangulation). A triangulation $\mathcal{T}$ of the point cloud $P = \{p_i : i = 1, \ldots, N\} \subset \mathbb{R}^2$ is a maximal family of triangles of the form

$$\mathcal{T} = \{T_i = \mathcal{C}(p_{i_1}, p_{i_2}, p_{i_3}) : \text{the edges } \overline{p_{i_1}p_{i_2}}, \overline{p_{i_1}p_{i_3}}, \overline{p_{i_2}p_{i_3}} \text{ do not intersect with any edge } \overline{p_{k_1}p_{k_2}}, \overline{p_{k_1}p_{k_3}}, \overline{p_{k_2}p_{k_3}} \text{ of any other triangle } T_k,\ k \neq i,\ i, k = 1, \ldots, M_T\}$$

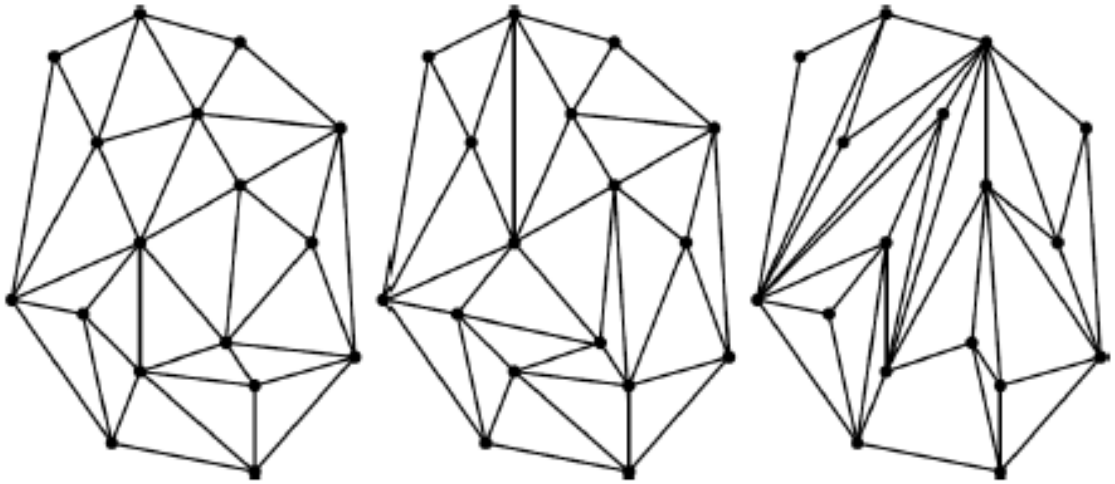
Maximality is meant in the sense that no triangle could be added to $\mathcal{T}$ without violating the 'no intersection'-property.

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 $p_l \in \mathrm{P}$ 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: $T_i \cap T_j = \emptyset \qquad \forall i \neq \mathrm{j}$

Triangulation

Triangulation of a point cloud is in general not unique.



Triangulation

Let $\mathcal{T}$ be a triangulation of the point cloud $P = \{p_i : i = 1, \ldots, N\}$, with all points being in general position. Then the triangulation contains

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

triangles and

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

edges. By K we denote the number of points in P which lie on the boundary of the convex hull $\mathcal{C}(P)$.

Triangulation

Let $\mathcal{T}$ be a triangulation of the point cloud $P = \{p_i : i = 1, \ldots, N\}$, with all points being in general position. Then the triangulation contains

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

triangles and

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

edges. By K we denote the number of points in P which lie on the boundary of the convex hull $\mathcal{C}(P)$.

Proof: This follows from Euler's formula for graphs.

The important aspect to remember from this: every triangulation of a given point cloud P has the **same amount of triangles and edges**.

Delaunay Triangulation

Let $\mathcal{D}(\mathcal{V})$ be the dual graph with respect to the Voronoi tesselation $\mathcal{V}$.

Definition (Delaunay Triangulation). A triangulation $\mathcal{T}$ is called a Delaunay triangulation if it is of the form

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

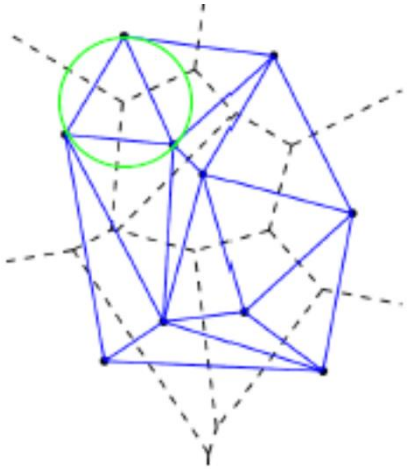
Delaunay Triangulation

Let $\mathcal{D}(\mathcal{V})$ be the dual graph with respect to the Voronoi tesselation $\mathcal{V}$.

Definition (Delaunay Triangulation). A triangulation $\mathcal{T}$ is called a Delaunay triangulation if it is of the form

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

Lemma. A Triangulation $\mathcal{T}$ of a pointcloud $P \subset \mathbb{R}^2$ is a Delaunay triangulation if and only if the circumcircle of every triangle $T \in \mathcal{T}$ does not contain any points of P in its interior.



Constructive Delaunay Triangulation

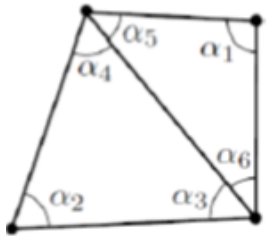
Definition (angle-maximal). Let $\mathcal{T}$ be a triangulation of a point cloud $P \subset \mathbb{R}^2$ and let $A(\mathcal{T}) = (\alpha_1, \alpha_2, \ldots, \alpha_{3M_T})$ be the sequence of angles in ascending order. Furthermore, let $A(\mathcal{T}') = (\alpha'_1, \alpha'_2, \ldots, \alpha'_{3M_T})$ be the sequence of angles in ascending oder for a second triangulation $\mathcal{T}'$ of P. Then we say

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

if there exists an index I such that

$$\alpha_i = \alpha'_i, \text{ if } i < I, \qquad \alpha_I > \alpha'_I.$$

The triangulation $\mathcal{T}$ is called angle-maximal if $A(\mathcal{T}) \geq A(\mathcal{T}')$ for every possible triangulation $\mathcal{T}'$ of P.

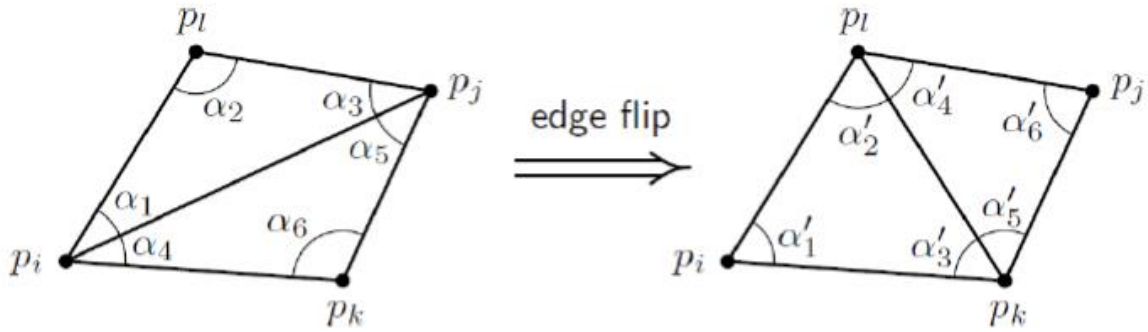


Constructive Delaunay Triangulation

Definition (illegal edge, edge flip). Let $\mathcal{T}$ be a triangulation of a point cloud $P \subset \mathbb{R}^2$ and let $T_1 = \mathcal{C}(p_i, p_j, p_l)$, $T_2 = \mathcal{C}(p_i, p_j, p_k)$ be two adjacent triangles of that triangulation, whose union forms a convex set. We define a second set of adjacent triangles by $T_1' = \mathcal{C}(p_i, p_k, p_l)$, $T_2' = \mathcal{C}(p_j, p_k, p_l)$. If it holds for the corresponding angles that

$$\min_{i=1,2,\ldots,6} \alpha_i < \min_{i=1,2,\ldots,6} \alpha_i',$$

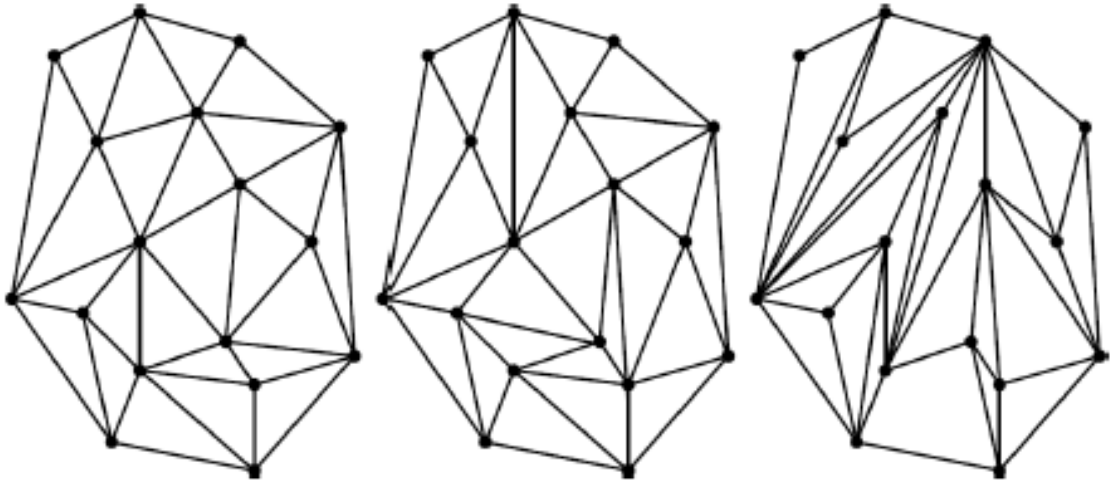
then we call $\overline{p_i p_j}$ an illegal edge. The transition from T_1, T_2 to T_1', T_2' is called edge flip.



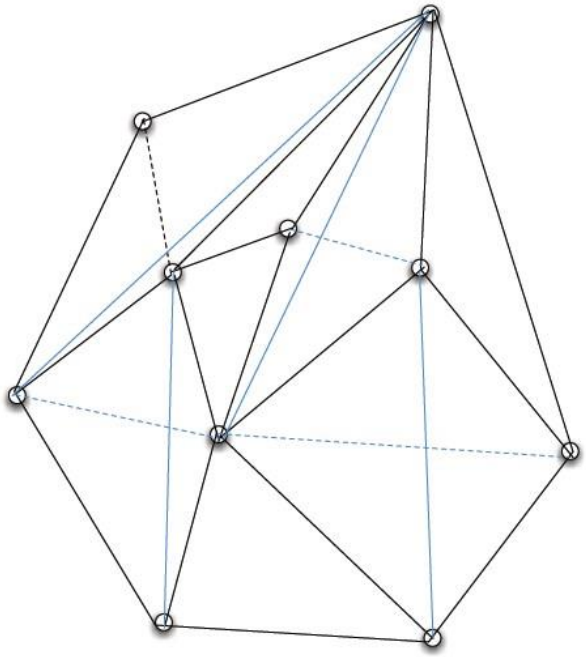
Constructive Delaunay Triangulation

If a triangulation $\mathcal{T}$ contains an illegal edge, then edge flipping leads to a triangulation $\mathcal{T}'$ with $A(\mathcal{T}') > A(\mathcal{T})$. Iteration until no more illegal edges exist leads to a Delaunay triangulation.

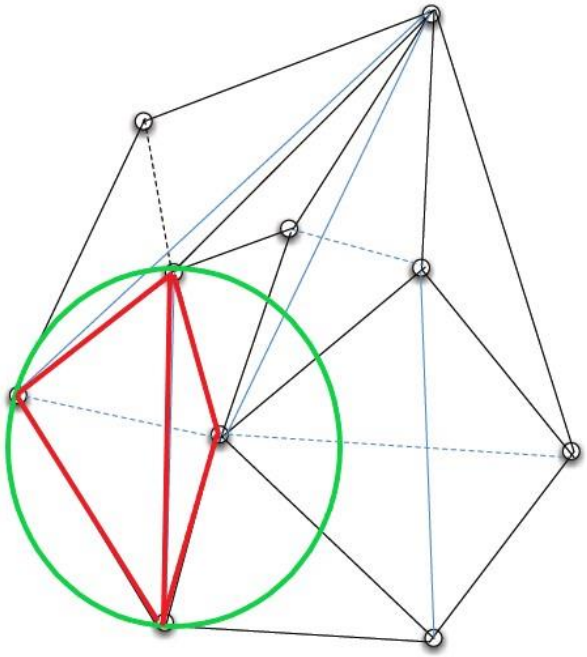
Theorem (Characterization of Delaunay triangulation). Let $\mathcal{T}$ be a triangulation of the point set $P \subset \mathbb{R}^2$. $\mathcal{T}$ is a Delaunay triangulation (in the sense of the definition via the dual of a Voronoi tesselation) if and only if it does not contain any illegal edges.



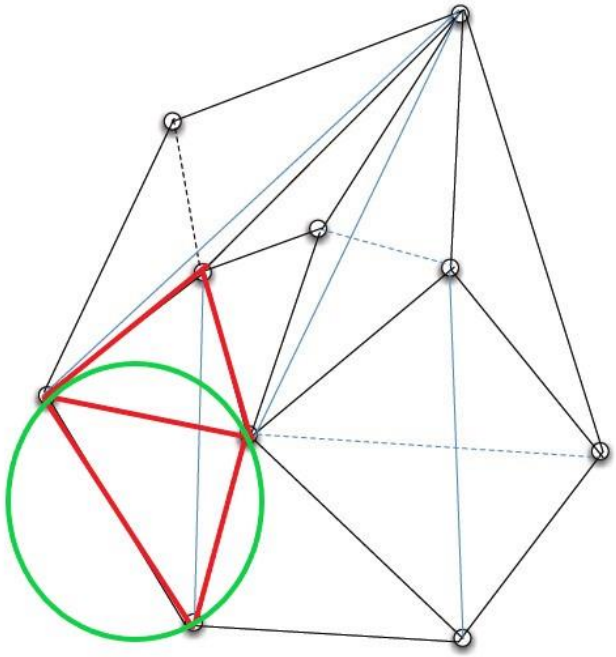
Constructive Delaunay Triangulation



Constructive Delaunay Triangulation



Constructive Delaunay Triangulation



Constrained Delaunay Triangulation

In some situations certain additional neighborhood information information can be available. For example:

- Known contour lines of equal altitude in a DEM (these contour lines should be maintained in a triangulation)
- Fault lines/discontinuities (edges alogn discontinuities should be maintained)

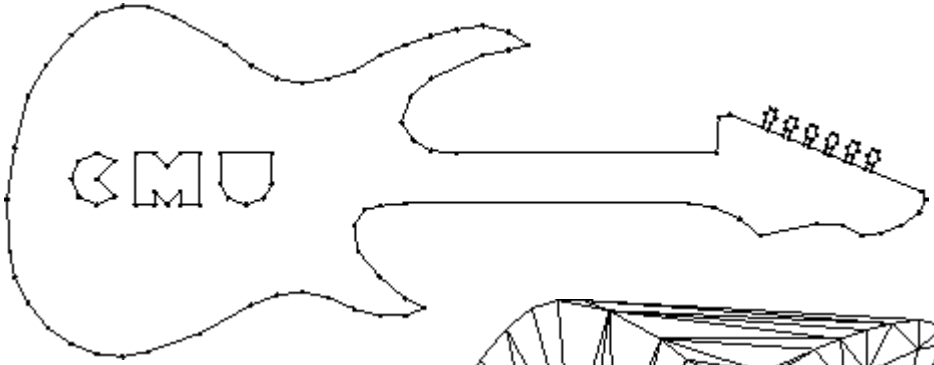
Constrained Delaunay Triangulation

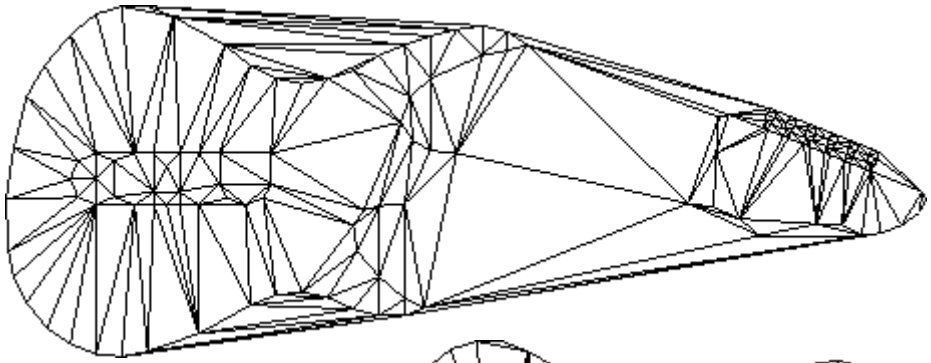
In some situations certain additional neighborhood information information can be available. For example:

- Known contour lines of equal altitude in a DEM (these contour lines should be maintained in a triangulation)
- Fault lines/discontinuities (edges along discontinuities should be maintained)

The **local character** of the construction of the Delaunay triangulation allows to include this prior knowledge as constraints. The resulting triangulation is locally optimal with respect to the max-min condition, except for those triangles that contain one of the constraint edges.

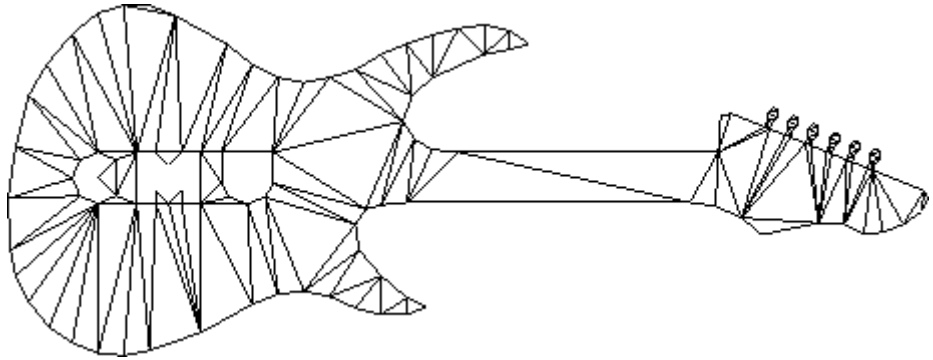
Constrained Delaunay Triangulation



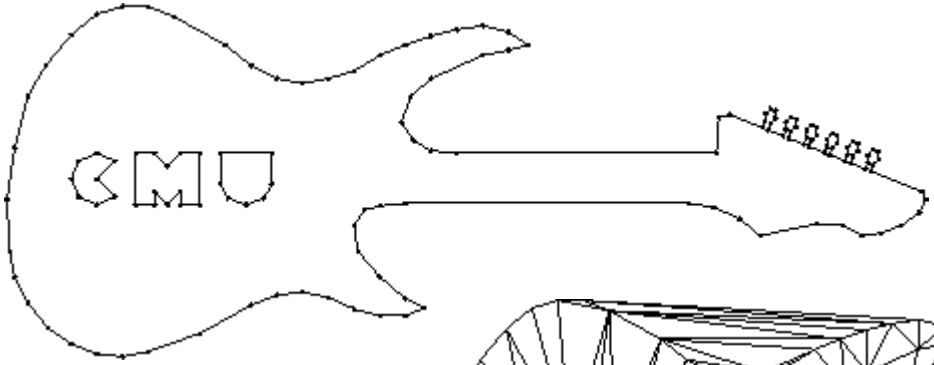


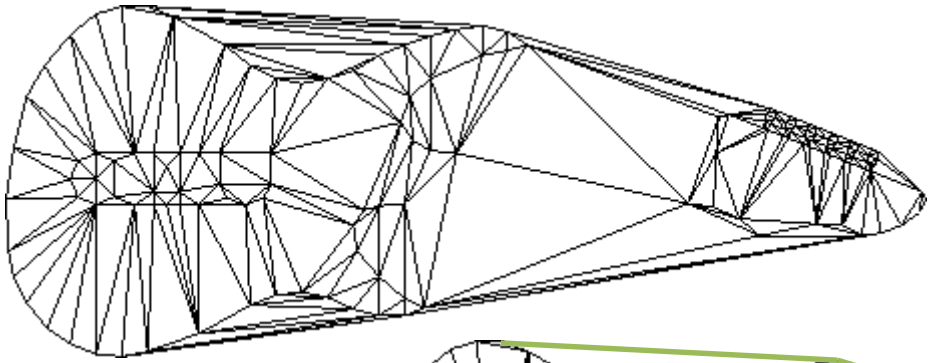
No constraints at all yield the following triangulation of the data points. Some features of the guitar get lost and we triangulate the entire convex hull.

Using the edges of the top image as constraints and removing all edges outside of the guitar frame yields the following triangulation.

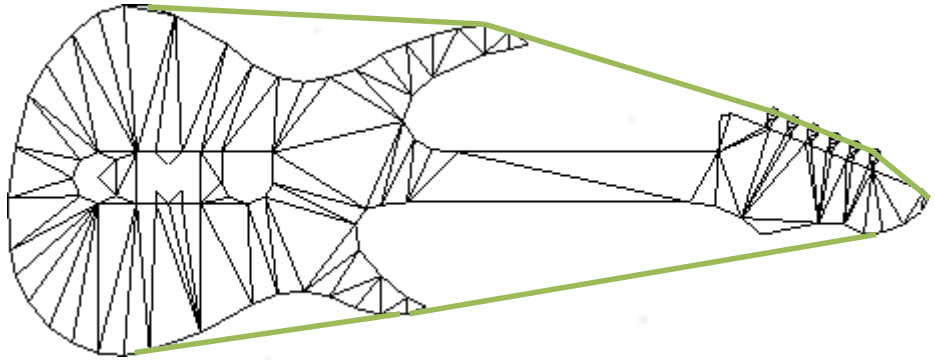


Constrained Delaunay Triangulation



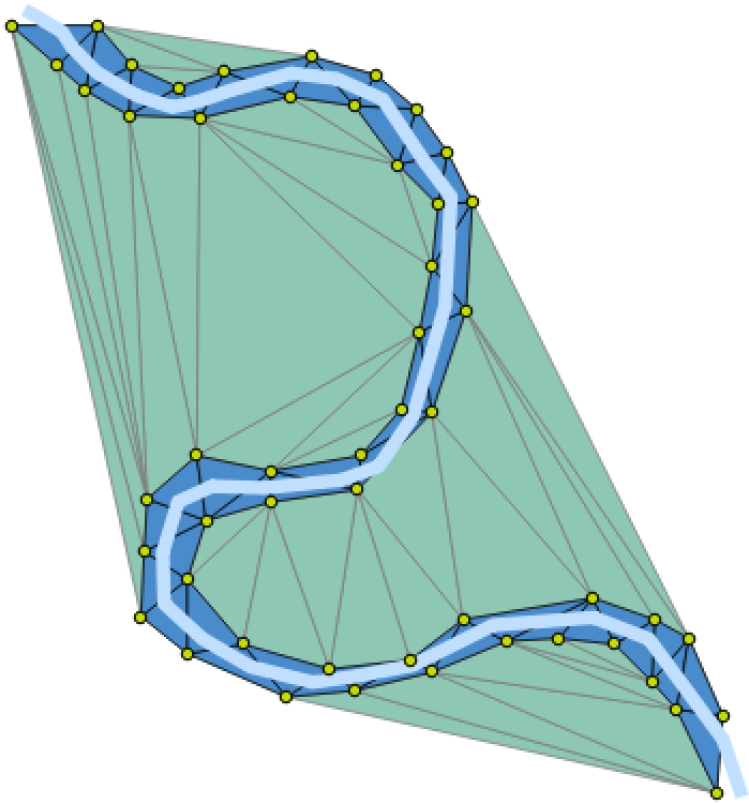


No constraints at all yield the following triangulation of the data points. Some features of the guitar get lost and we triangulate the entire convex hull.



Using the edges of the top image as constraints and removing all edges outside of the guitar frame yields the following triangulation.

Constrained Delaunay Triangulation



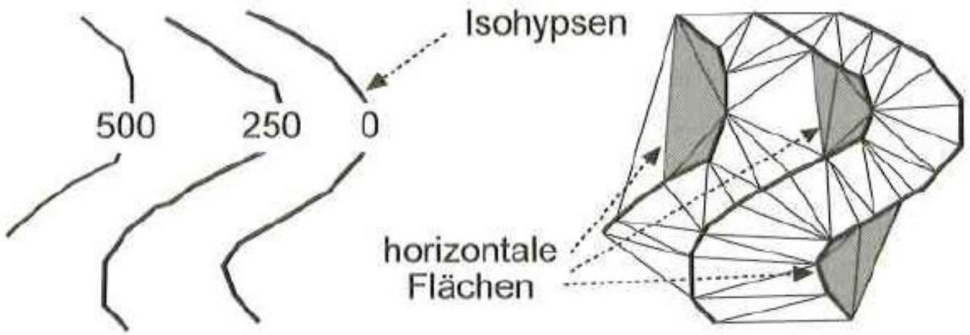
Another example: Triangulation constrained by edges describing the river bank.



Triangulation constrained by edges describing the coastlines of an island. Additonally, all edges outside of the land area are removed.

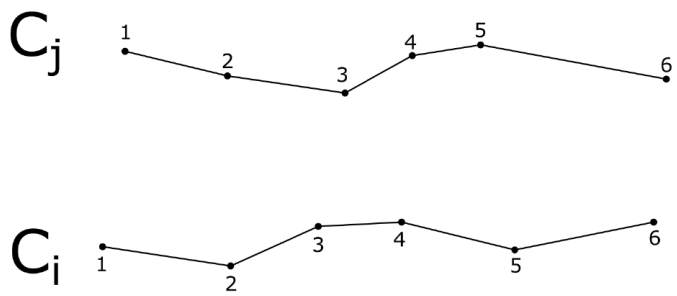
Parkettierung

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



Parkettierung

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

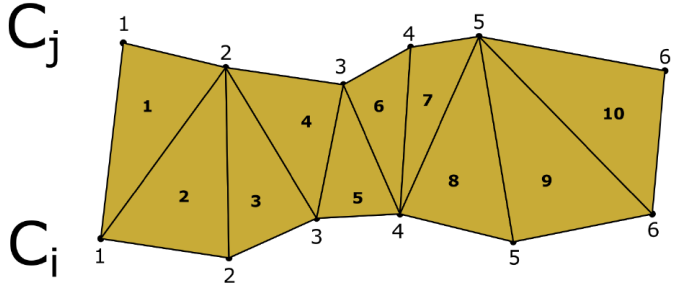


Es seien zwei Folgen $C_i = \{p_1, \ldots, p_n\}$ und $C_j = \{q_1, \ldots, q_m\}$ von Punkten gegeben.
Eine **Parkettierung (Tiling)** mittels Dreiecken ist dann gegeben, wenn folgende Bedingungen erfüllt sind:

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.

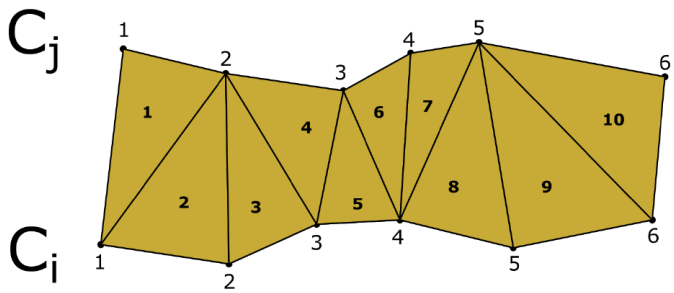
Parkettierung

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

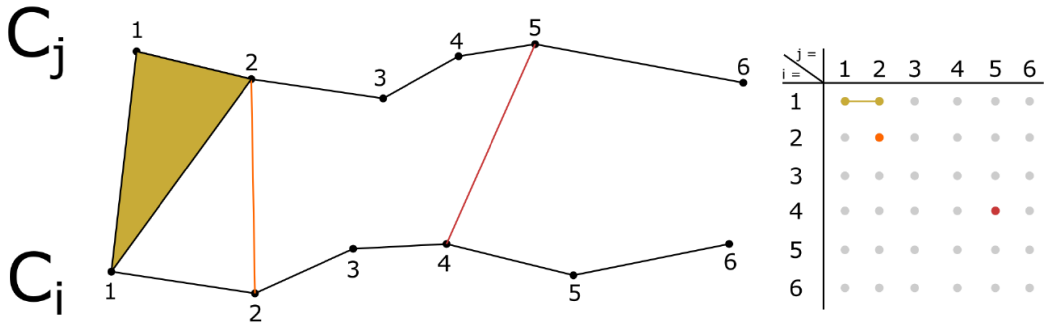


Parkettierung

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

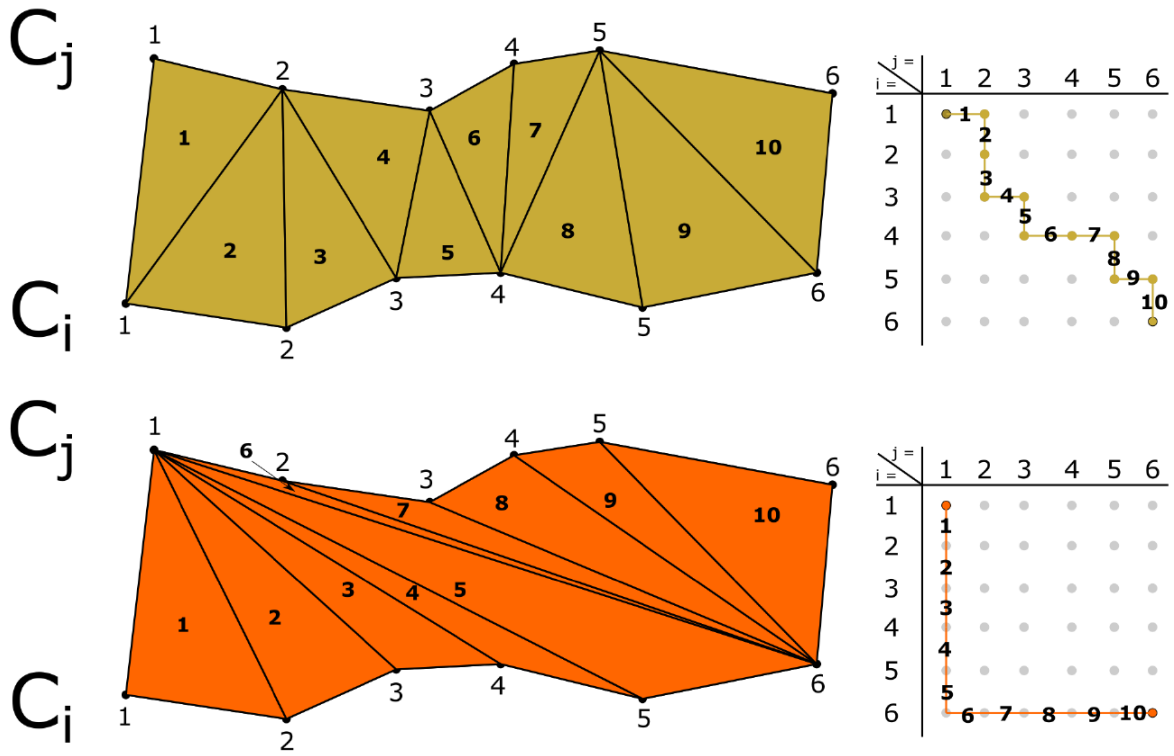


Parkettierungen können mit Hilfe eines Suchgraphens dargestellt werden: jeder Eintrag der Matrix beschreibt eine Verbindungslinie zwischen den Punkten der jeweiligen Zeile und Spalte.



Parkettierung

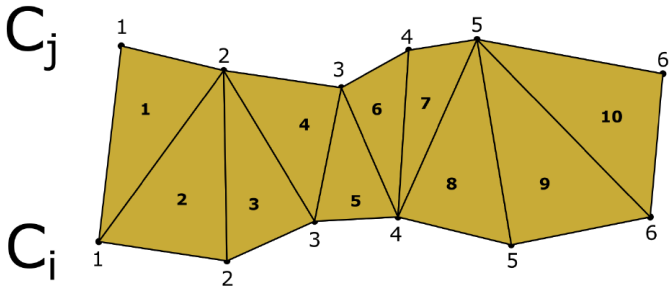
Jede Parkettierung kann mittels eines Pfades in dem Suchgraphen dargestellte werden



Parkettierung

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

Dazu stellt man G zeln gewichtet, als auch kumulativ gewichtet



Gewichte

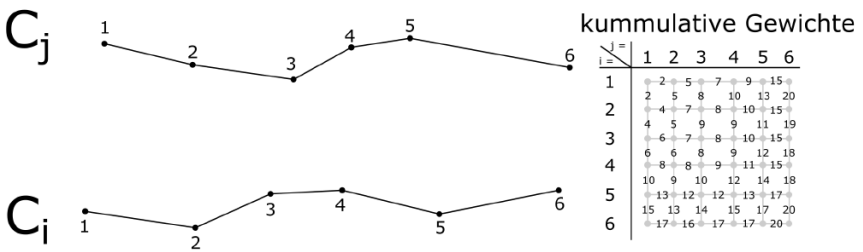
j = / i =	1	2	3	4	5	6
1	2	3	2	2	6	
2	2	3	3	3	4	5
3	2	3	1	2	5	
4	2	1	2	1	1	4
5	2	2	1	2	5	
6	2	1	1	1	2	3
	2	2	1	2	4	
	4	3	2	3	3	3
	3	3	2	1	4	
	5	4	4	3	4	3
	2	3	3	2	3	

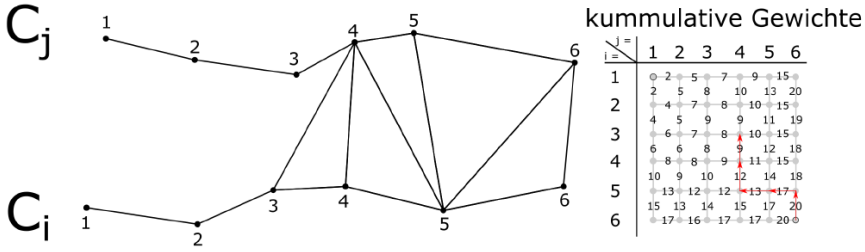
kummulative Gewichte

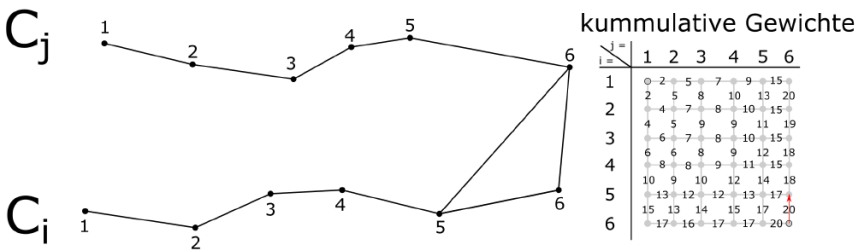
j = / i =	1	2	3	4	5	6
1	2	5	7	9	15	
2	2	5	8	10	13	20
3	4	7	8	10	15	
4	4	5	9	9	11	19
5	6	7	8	10	15	
6	6	6	8	9	12	18
	8	8	9	11	15	
	10	9	10	12	14	18
	13	12	12	13	17	
	15	13	14	15	17	20
	17	16	17	17	20	

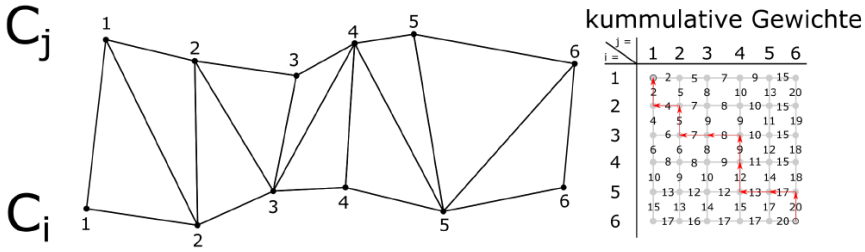
Parkettierung

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





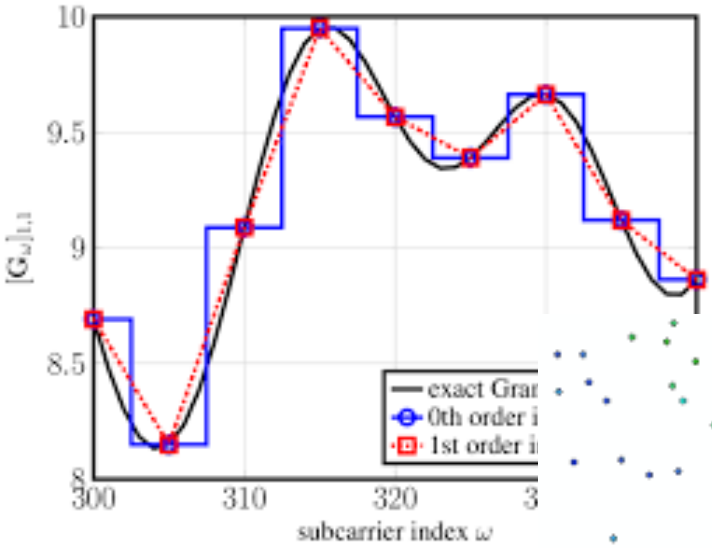


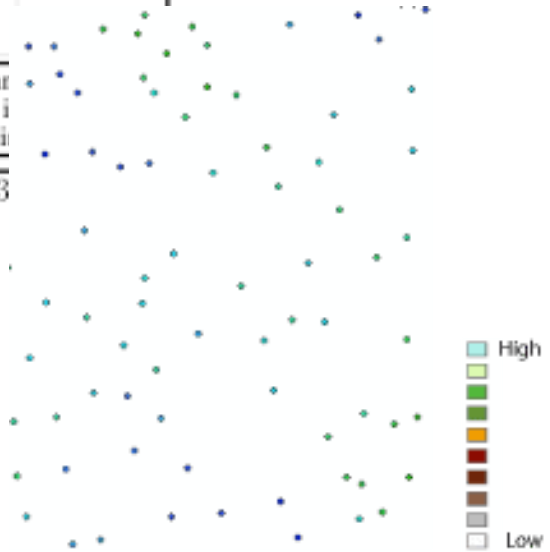


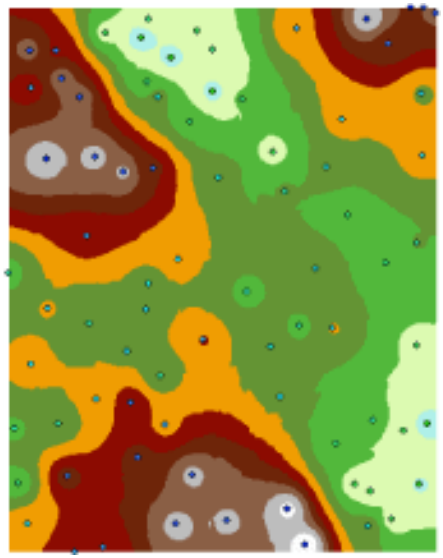
Übersicht

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

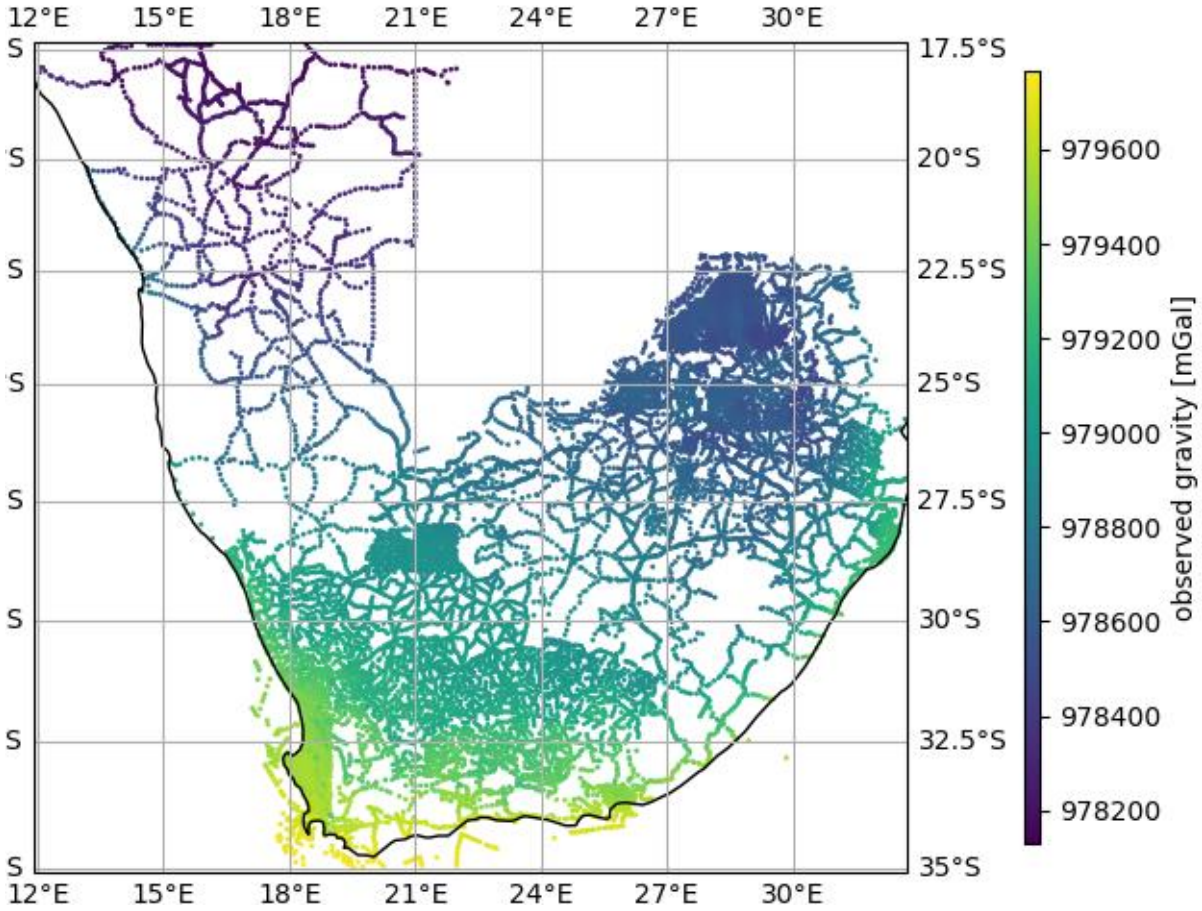
Examples


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


High
Low

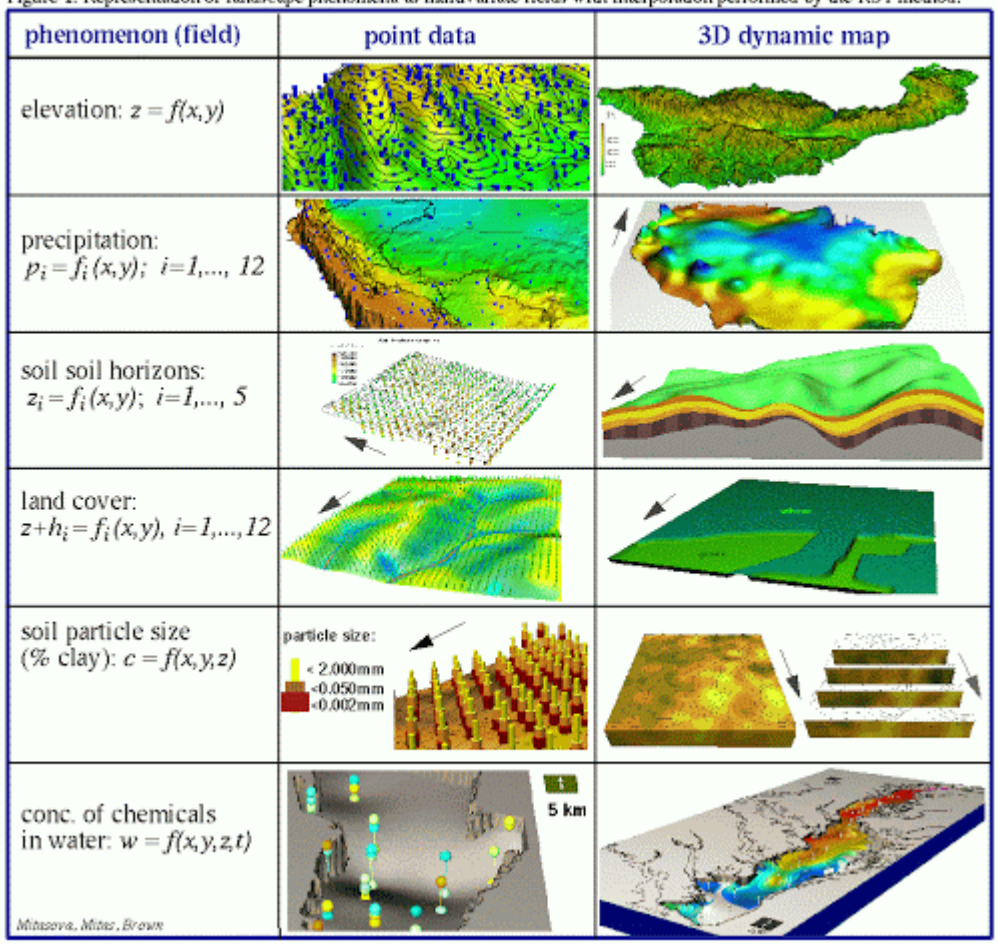


Examples


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]

Examples

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

What is Interpolation?

Typically measurements are available in only finitely many locations $x_1, \ldots, x_N$ (these may be physical locations in space, points in time, or abstract locations). At each location x_i we assume to have one measurement y_i of a quantity of interest (more than one measurement might be possible as well, but will not be considered here).

The goal of interpolation is to "predict" the quantity of interest y at a location x where no measurements are available.

This typically amounts to finding a function f such that

$$f(x_i) = y_i, \qquad i = 1, \ldots, N,$$

with adequate properties. The "prediction" at location x is the given by $y = f(x)$.

What is Interpolation?

The interpolating function will typically represented in one of the two forms

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

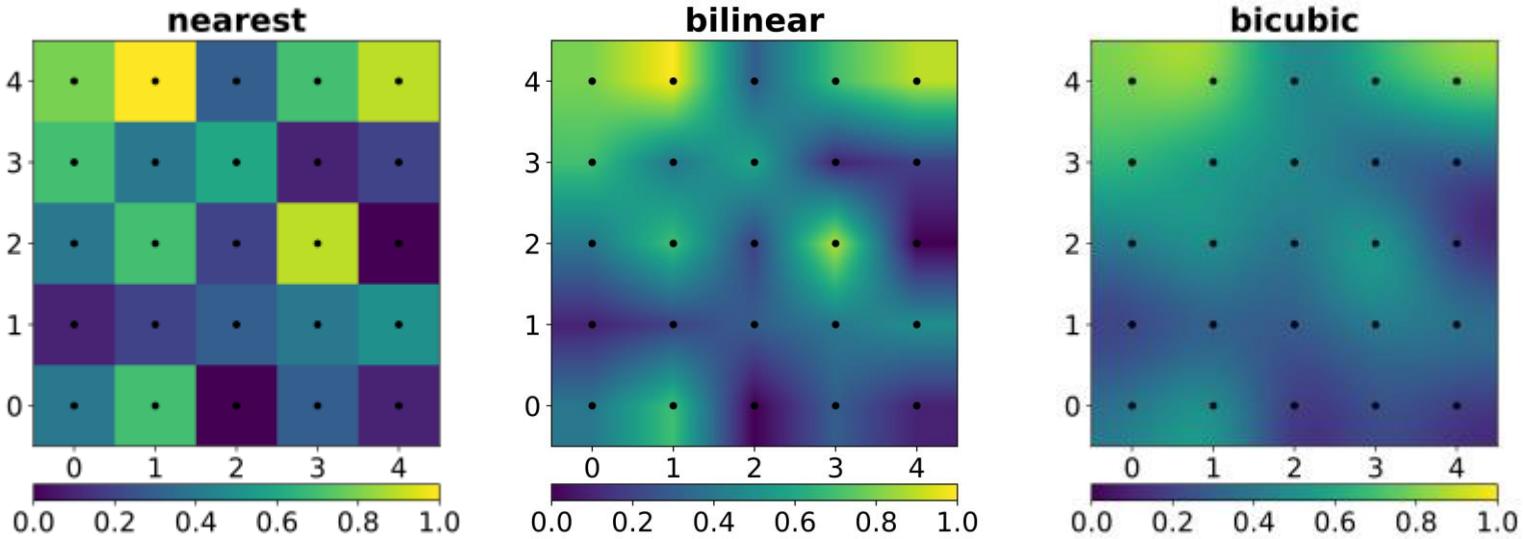
with weights λ_i depending on the location x, or

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

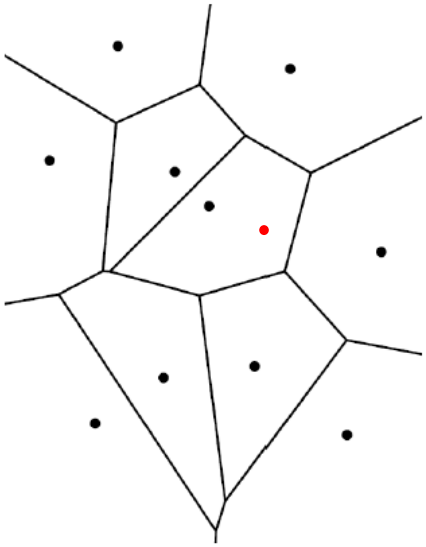
with fixed weights λ_i (depending on the measurements y_i).

Grid based and Element Based Methods

- **Nearets Neighbour Interpolation** (Voronoi Tesselation required): discontinuous
- **Linear Interpolation on Triangulations**: continuous
- **Natural Neighbour Interpolation** (Voronoi Tesselation required): continuously differentiable
- **Bilinear Interpolation** (rectilinear grid required)



Nearest Neighbour Interpolation



Let $\mathcal{V} = \{V_1, \ldots, V_m\}$ be the Voronoi tesselation of $P = \{p_1, \ldots, p_m\} \subset \mathbb{R}^n$. If y_i is the value assigned to/measured at p_i, then we assign

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

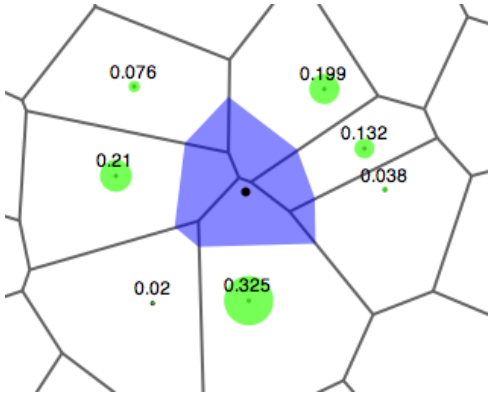
with weights

$$\lambda_i(x) = \begin{cases} 1, & \text{if } x \in V_i, \\ 0, & \text{else.} \end{cases}$$

Properties:

- $f(x)$ is discontinuous
- local interpolator that relies **only** on (p_i, y_i) if $x \in V_i$
- extrapolation is technically possible

Natural Neighbour Interpolation



Let $\mathcal{V}$ be the Voronoi tesselation of $P = \{p_1, \ldots, p_m\} \subset \mathbb{R}^n$. Let $x \in \mathbb{R}^n$ an arbitrary point and $\mathcal{V}'$ the Voronoi tesselation with respect to $P \cup \{x\}$. If p_{i_x}, $i = 1, \ldots, n_x$ are the strong Voronoi neighbours of x with respect to $\mathcal{V}'$, then we set

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

If y_i is the value assigned to/measured at p_i, then we assign

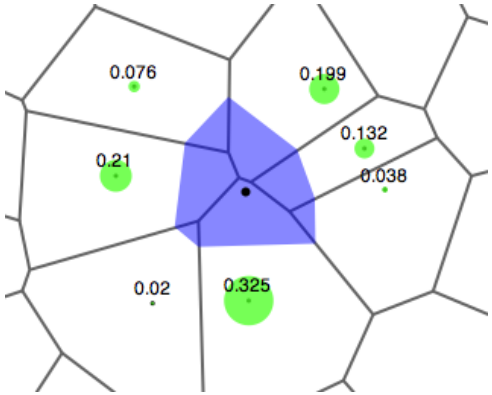
$$f(x) = \sum_{i=1}^{n_x} \lambda_{i_x} y_{i_x}.$$

The $\lambda_{i_x} > 0$ satisfy

$$\sum_{i=1}^{n_x} \lambda_{i_x} = 1$$

This is a typical property of weights in for interpolation. (Isoparametric property)

Natural Neighbour Interpolation



Let $\mathcal{V}$ be the Voronoi tesselation of $P = \{p_1, \ldots, p_m\} \subset \mathbb{R}^n$. Let $x \in \mathbb{R}^n$ an arbitrary point and $\mathcal{V}'$ the Voronoi tesselation with respect to $P \cup \{x\}$. If p_{i_x}, $i = 1, \ldots, n_x$ are the strong Voronoi neighbours of x with respect to $\mathcal{V}'$, then we set

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

If y_i is the value assigned to/measured at p_i, then we assign

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

Properties:

- $f(x)$ is continuous and differentiable
- Local interpolator that relies **only** on all (p_i, y_i) that are the natural neighbours of x
- extrapolation is technically possible

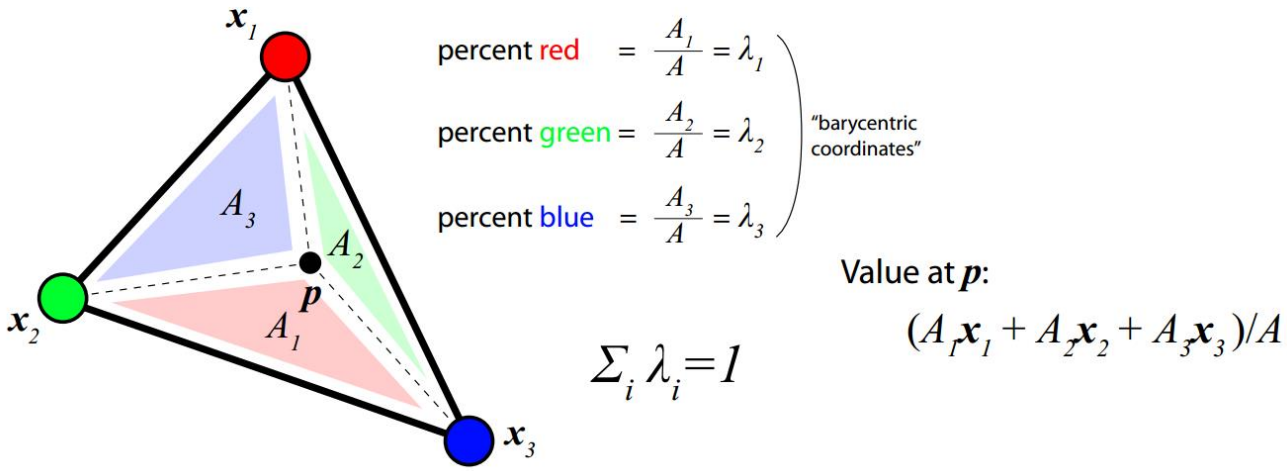
Linear Interpolation on Triangulations

Let x be a point inside the triangle $\mathcal{C}(x_i, x_j, x_k)$, with data y_i, y_j, y_k being available in the points x_i, x_j, x_k. If the point is represented in the form

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

then the linearly interpolated value in x is defined as

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



Linear Interpolation on Triangulations

Let x be a point inside the triangle $\mathcal{C}(x_i, x_j, x_k)$, with data y_i, y_j, y_k being available in the points x_i, x_j, x_k. If the point is represented in the form

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

then the linearly interpolated value in x is defined as

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

The coefficients $\lambda_i, \lambda_j, \lambda_k$ are denoted as **barycentric coordinates** of x (in 2-D). If $x = (x^{(1)}, x^{(2)})$ then they can be expressed as

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

with matrices

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

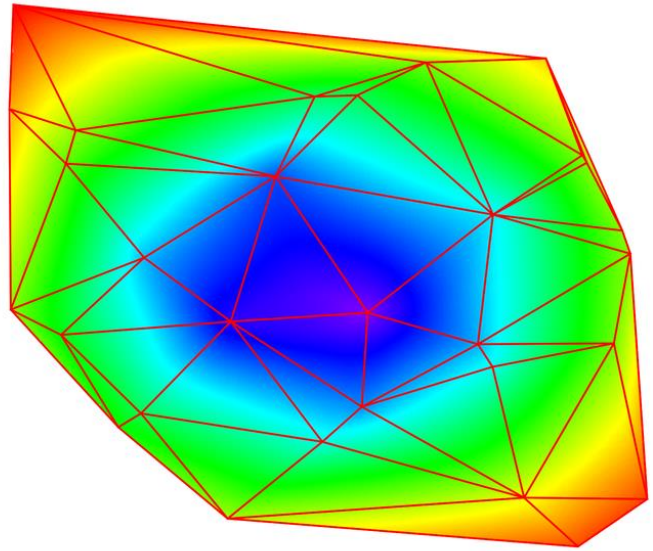
Linear Interpolation on Triangulations

Let x be a point inside the triangle $\mathcal{C}(x_i, x_j, x_k)$, with data y_i, y_j, y_k being available in the points x_i, x_j, x_k. If the point is represented in the form

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

then the linearly interpolated value in x is defined as

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



Linear Interpolation on Triangulations

Let x be a point inside the triangle $\mathcal{C}(x_i, x_j, x_k)$, with data y_i, y_j, y_k being available in the points x_i, x_j, x_k. If the point is represented in the form

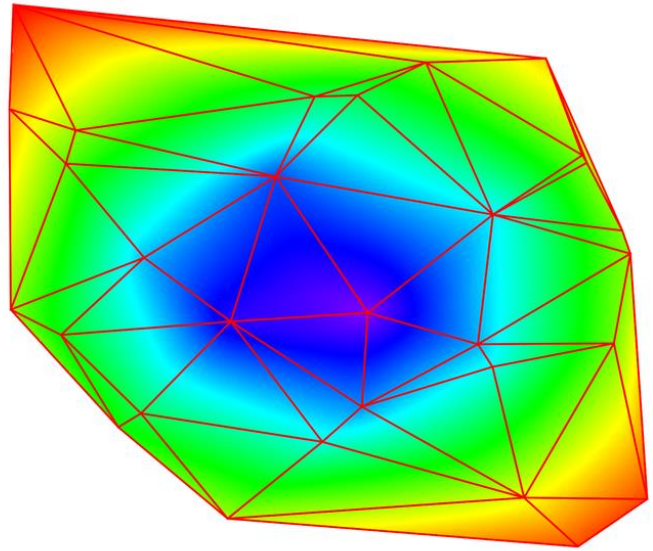
$$x = \lambda_i x_i + \lambda_j x_j + \lambda_k x_k,$$

then the linearly interpolated value in x is defined as

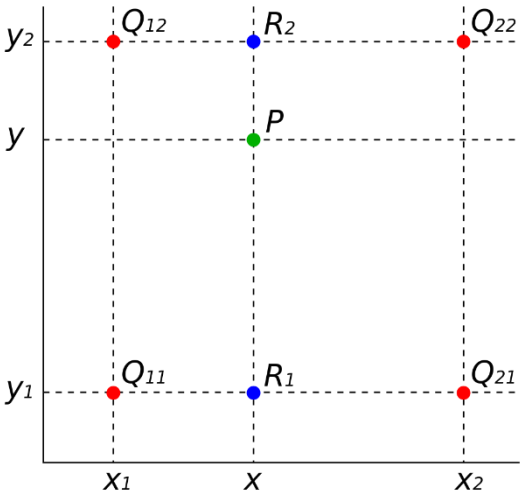
$$f(x) = \lambda_i y_i + \lambda_j y_j + \lambda_k y_k.$$

Properties:

- $f(x)$ is continuous
- over triangle edges $f(x)$ is not differentiable
- local interpolator that relies **only** on $(x_i y_i), (x_j, y_j), (x_k, y_k)$ if $x \in \mathcal{C}(x_i, x_j, x_k)$
- extrapolation is **not** initially possible



Bilinear Interpolation



If values $y_{i,j}$ are given in the points $Q_{i,j} \in \mathbb{R}^2$, then set in a point $P = (x^{(1)}, x^{(2)}) \in \mathbb{R}^2$ located in the rectangle defined by the points $Q_{i,j}, Q_{i+1,j}, Q_{i,j+1}, Q_{i+1,j+1}$:

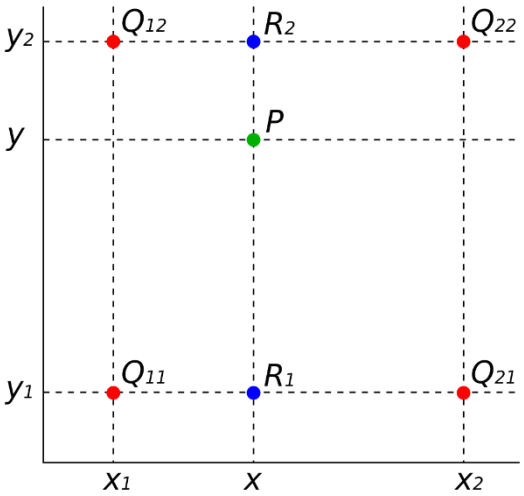
$$f(x^{(1)}, x^{(2)}) = \frac{x^{(2)}_{i,j+1} - x^{(2)}}{x^{(2)}_{i,j+1} - x^{(2)}_{i,j}} f_{i,j} + \frac{x^{(2)} - x^{(2)}_{i,j}}{x^{(2)}_{i,j+1} - x^{(2)}_{i,j}} f_{i,j+1}$$

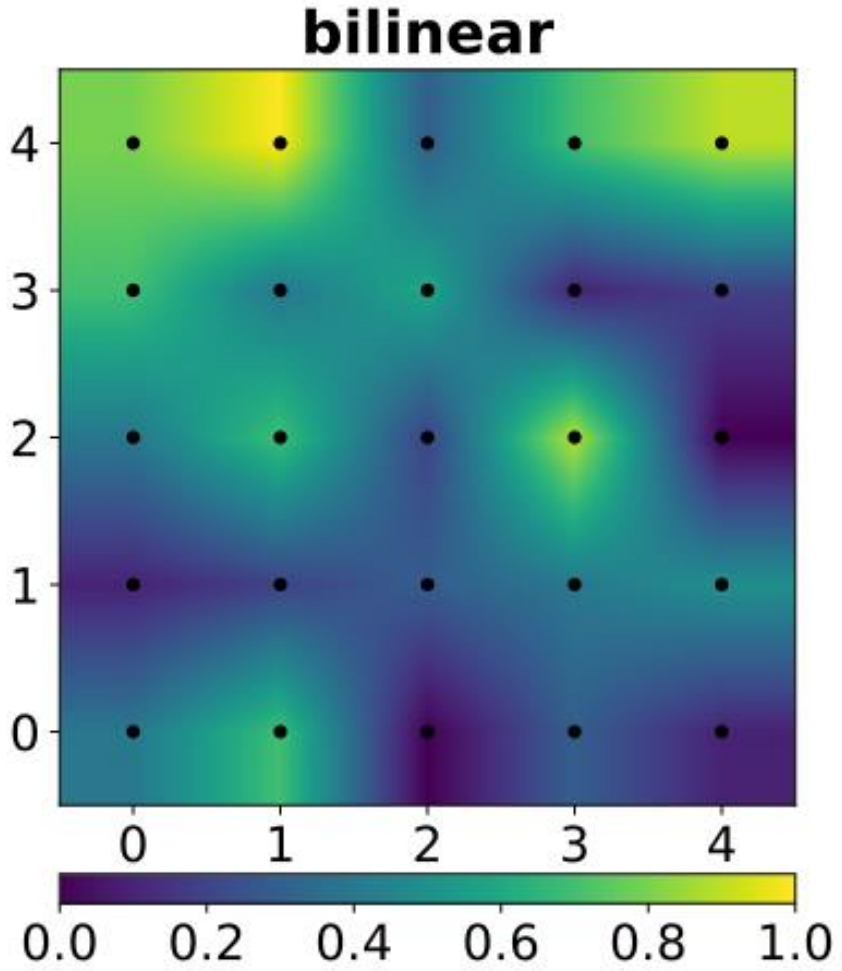
with

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

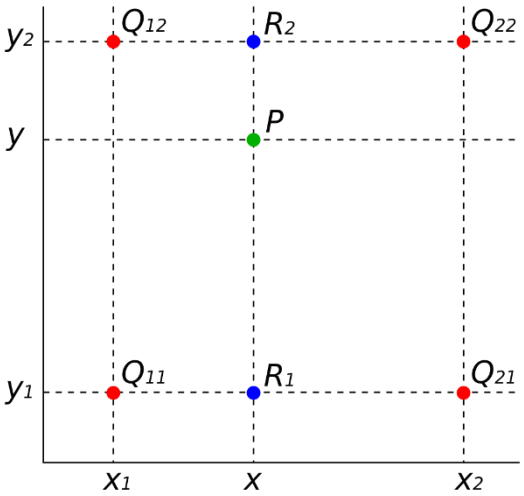
$$f_{i,j+1} = \frac{x^{(1)}_{i+1,j} - x^{(1)}}{x^{(1)}_{i+1,j} - x^{(1)}_{i,j}} y_{i,j+1} + \frac{x^{(1)} - x^{(1)}_{i,j}}{x^{(1)}_{i+1,j} - x^{(1)}_{i,j}} y_{i+1,j+1}$$

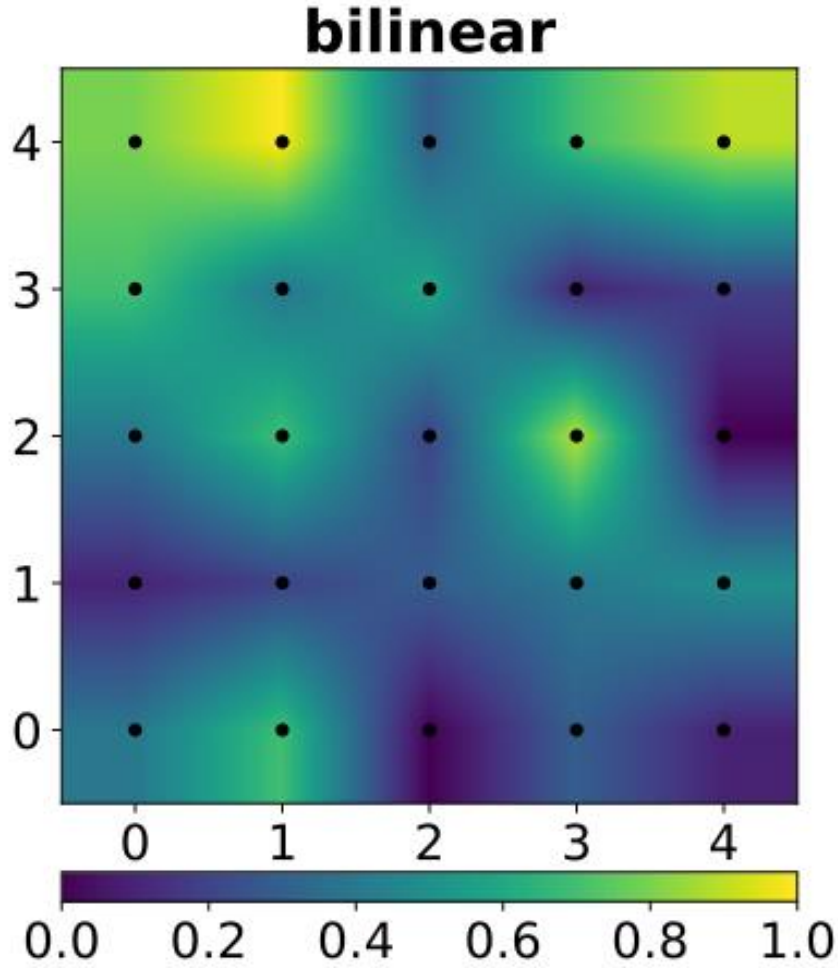
Bilinear Interpolation





Bilinear Interpolation

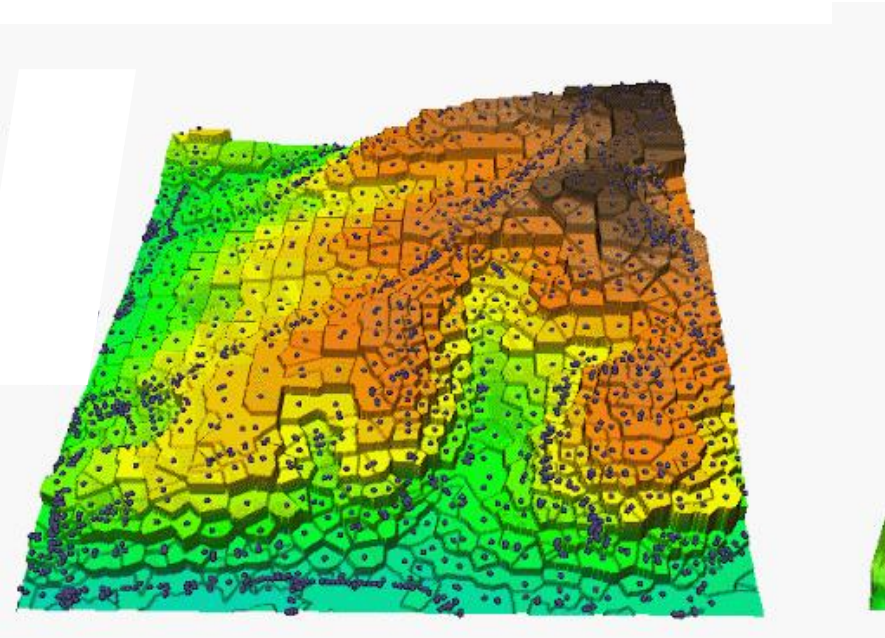


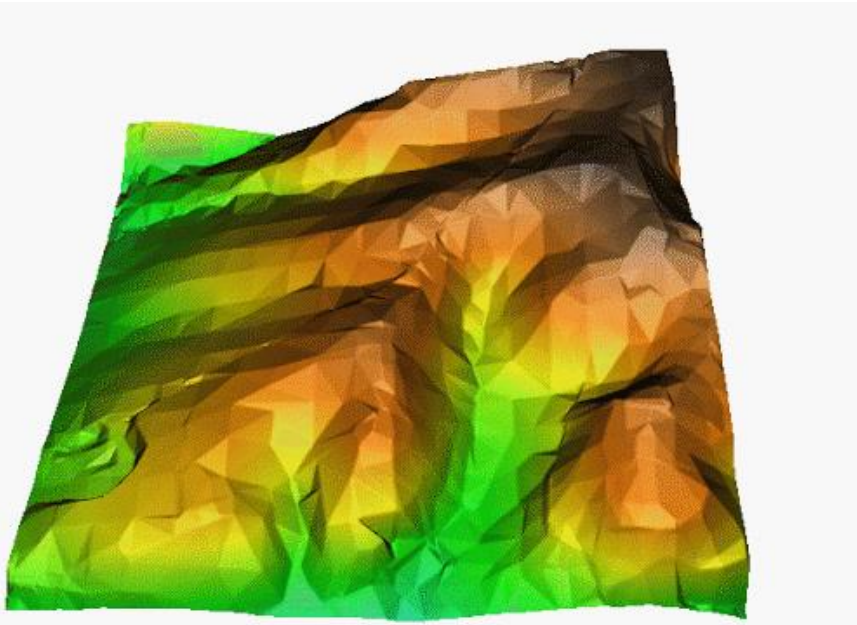


Properties:

- $f(x)$ is continuous
- over rectangle edges $f(x)$ is not differentiable
- Local interpolator that relies **only** on $(x_{i,j}y_{i,j}), (x_{i+1,j}, y_{i+1,j}), (x_{i,j+1}, y_{i,j+1}), (x_{i+1,j+1}, y_{i+1,j+1})$ if x is inside the rectangle
- extrapolation is **not** initially possible

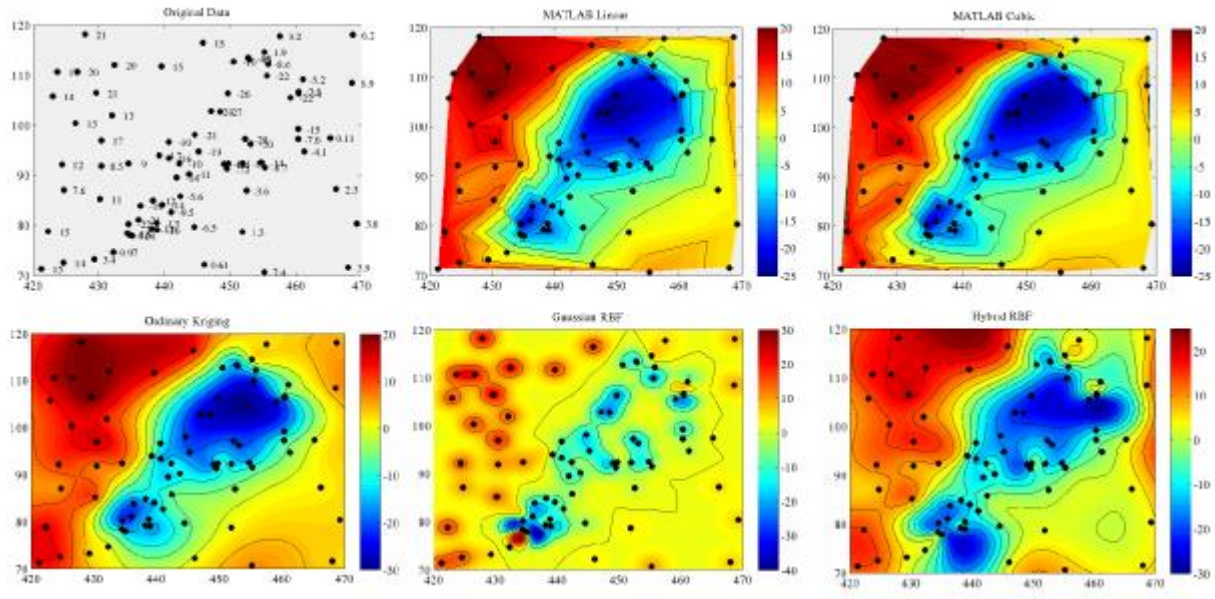
Example Nearest Neighbour, Lin. Interp. on Triangulation





Grid free Methods

- **Inverse distance weighting**
- **Classical Polynomial Interpolation**
- **Splines and Radial Basis Functions**



Inverse Distance Weighting (IDW)

If values y_i are given in the points $x_1, \ldots, x_N \in \mathbb{R}^n$, then set in a point $x \in \mathbb{R}^n$:

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

with

$$\lambda_i(x) = \frac{\frac{1}{|x-x_i|^k}}{\sum_{j=1}^{N} \frac{1}{|x-x_j|^k}} = \frac{1}{1 + \sum_{j=1, j\neq i}^{N} \frac{|x-x_i|^k}{|x-x_j|^k}}$$

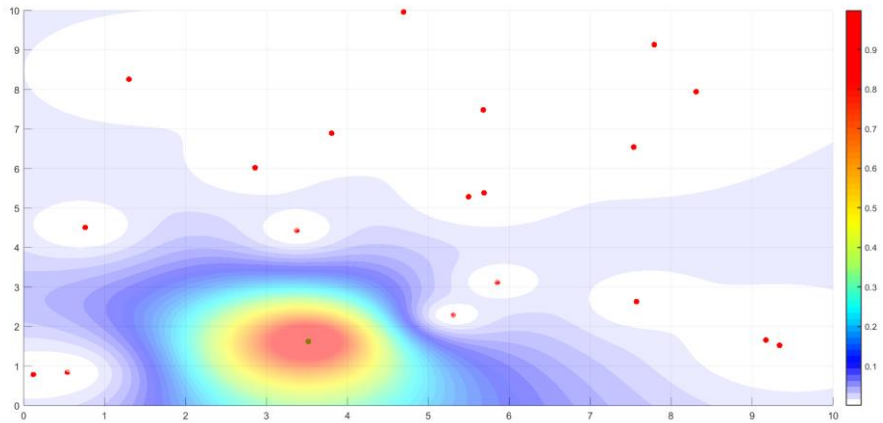


Illustration of weight function λ_1 for various $x \in [0,1]^2$ with parameter $k = 2$.

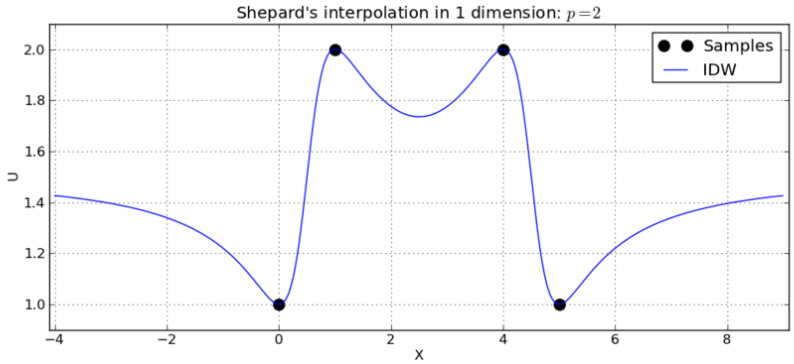
Inverse Distance Weighting (IDW)

If values y_i are given in the points $x_1, \ldots, x_N \in \mathbb{R}^n$, then set in a point $x \in \mathbb{R}^n$:

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

with

$$\lambda_i(x) = \frac{\frac{1}{|x-x_i|^k}}{\sum_{j=1}^{N} \frac{1}{|x-x_j|^k}} = \frac{1}{1 + \sum_{j=1, j\neq i}^{N} \frac{|x-x_i|^k}{|x-x_j|^k}}$$



Interpolation example 1-D.

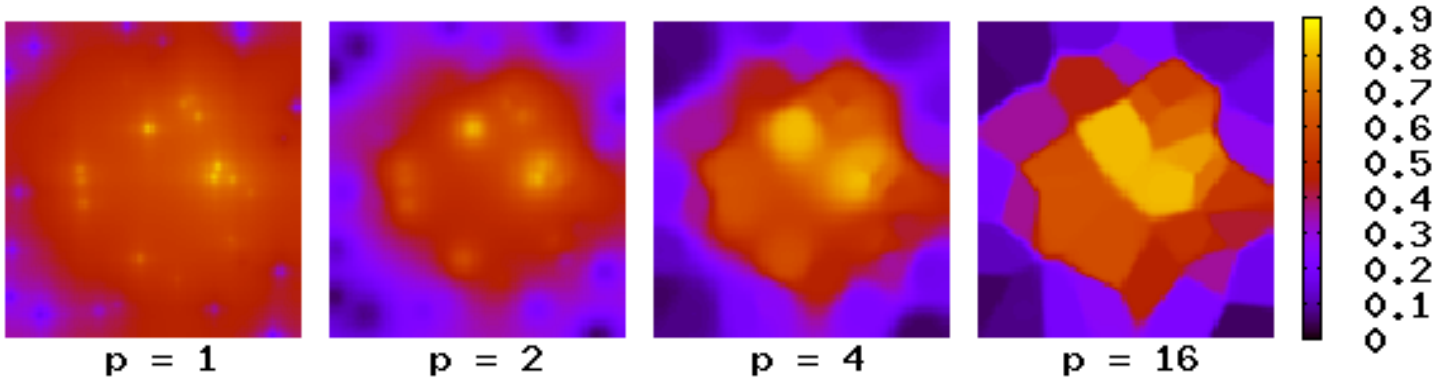
Inverse Distance Weighting (IDW)

If values y_i are given in the points $x_1, \ldots, x_N \in \mathbb{R}^n$, then set in a point $x \in \mathbb{R}^n$:

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

with

$$\lambda_i(x) = \frac{\frac{1}{|x-x_i|^k}}{\sum_{j=1}^{N} \frac{1}{|x-x_j|^k}} = \frac{1}{1 + \sum_{j=1, j\neq i}^{N} \frac{|x-x_i|^k}{|x-x_j|^k}}$$



Interpolation example 2-D.

Inverse Distance Weighting (IDW)

If values y_i are given in the points $x_1, \ldots, x_N \in \mathbb{R}^n$, then set in a point $x \in \mathbb{R}^n$:

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

with

$$\lambda_i(x) = \frac{\frac{1}{|x-x_i|^k}}{\sum_{j=1}^{N} \frac{1}{|x-x_j|^k}} = \frac{1}{1 + \sum_{j=1, j\neq i}^{N} \frac{|x-x_i|^k}{|x-x_j|^k}}$$

Properties:

- $f(x)$ is continuous and differentiable
- global interpolator that relies on **all** (x_i, y_i) with $1 \leq i \leq N$
- weights λ_i rely only on inter-point-distances
- extrapolation is technically possible

Polynomial Interpolation

If values y_i are given in the points $x_1, \ldots, x_N \in \mathbb{R}$ and $f(x) = \sum_{i=0}^{m} \lambda_i x^i$ is a polynomial of degree $m = N - 1$, then choose the coefficients $\lambda_0, \ldots, \lambda_{n-1} \in \mathbb{R}$ such that

$$f(x_i) = y_i, \quad i = 1, \ldots, N.$$

Finding the coefficients $\lambda_0, \ldots, \lambda_m$ corresponds to solving the following system of linear equations:

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

with $\boldsymbol{\lambda} = (\lambda_0, \ldots, \lambda_m)^T$, $\mathbf{y} = (y_1, \ldots, y_N)^T$ and

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

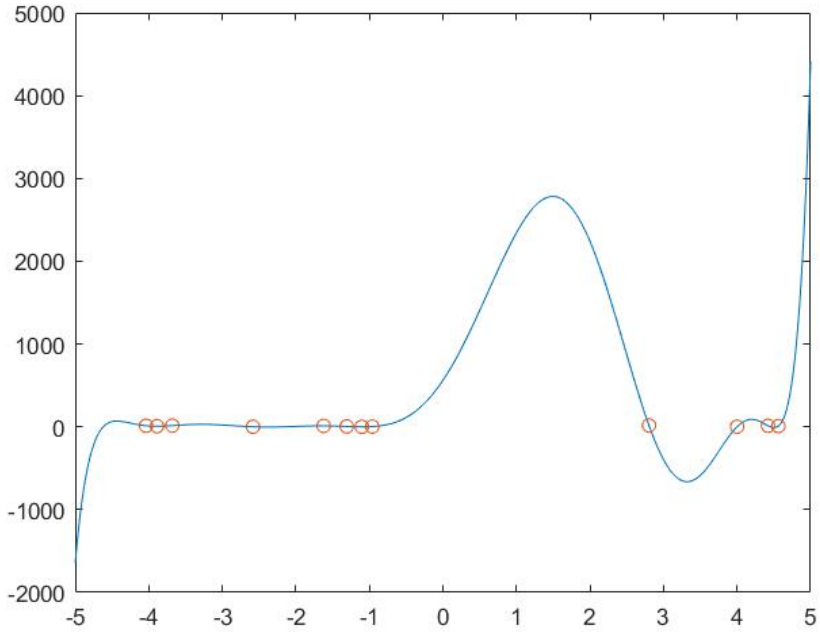
Polynomial Interpolation

If values y_i are given in the points $x_1, \ldots, x_N \in \mathbb{R}$ and $f(x) = \sum_{i=0}^{m} \lambda_i x^i$ is a polynomial of degree $m = N - 1$, then choose the coefficients $\lambda_0, \ldots, \lambda_{n-1} \in \mathbb{R}$ such that

$$f(x_i) = y_i, \quad i = 1, \ldots, N.$$

Finding the coeffic : linear equations:

with $\boldsymbol{\lambda} = (\lambda_0, \ldots,$



Polynomial Interpolation with N=12 data points and polynomial degree m=N-1=11.

Polynomial Interpolation

Let values y_i be given in the points $x_1, \ldots, x_N \in \mathbb{R}$ and $f(x) = \sum_{i=0}^{m} \lambda_i x^i$ be a polynomial of degree m. Then the oscillation effect of polynomial interpolation can be reduced if we choose $m \ll N - 1$ and choose the coefficients $\lambda_0, \ldots, \lambda_m$ by minimizing

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

In the case we talk about **approximation** (or occasionally in-exact inteprolation) because we can typically not achieve $f(x_i) = y_i$ for all $i = 1, \ldots, N$ but only

$$f(x_i) \approx y_i, \qquad i = 1, \ldots, N.$$

Properties:

- $f(x)$ is a polynomial of degree m (therefore continuous and m-times differentiable)
- global interpolator that relies on **all** (x_i, y_i) with $1 \leq i \leq N$
- if $m \ll N - 1$: approximation-polynomial with $f(x_i) \approx y_i$ with $1 \leq i \leq N$
- extrapolation is technically possible

Splines 1-D

A **spline of degree** n is a function that, for given nodes $P = \{x_1, \ldots, x_N\} \subset \mathbb{R}$, coincides with polynomials of degree n on each open interval (x_i, x_{i+1}), (x_n, ∞), and $(-\infty, x_1)$ and whose derivatives up to order $n-1$ are continuous at all points $x_1, \ldots, x_N$.

We denote the space of functions comprising such splines by $\mathrm{SP}_{P,n}$.

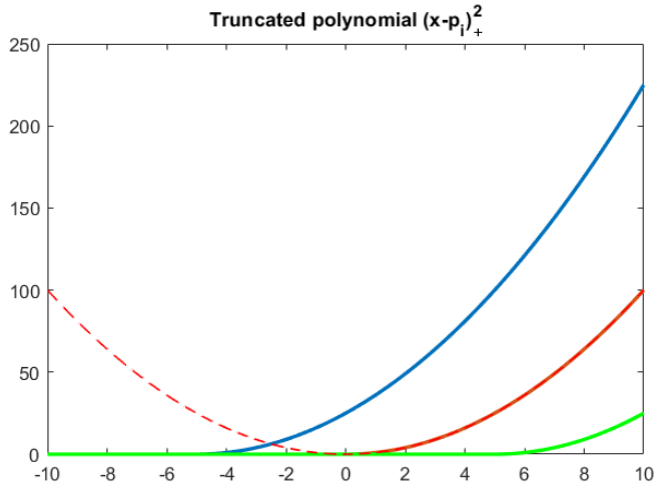
Splines 1-D

A **spline of degree** n is a function that, for given nodes $P = \{x_1, \ldots, x_N\} \subset \mathbb{R}$, coincides with polynomials of degree n on each open interval (x_i, x_{i+1}), (x_n, ∞), and $(-\infty, x_1)$ and whose derivatives up to order $n-1$ are continuous at all points $x_1, \ldots, x_N$.

We denote the space of functions comprising such splines by $\mathrm{SP}_{P,n}$.

Examples of elements of $\mathrm{SP}_{P,n}$ are the truncated polynomials

$$(x - p_i)_+^n = \begin{cases} (x-p_i)^n, & \text{if } x \geq p_i, \\ 0, & \text{else.} \end{cases}$$



Splines 1-D

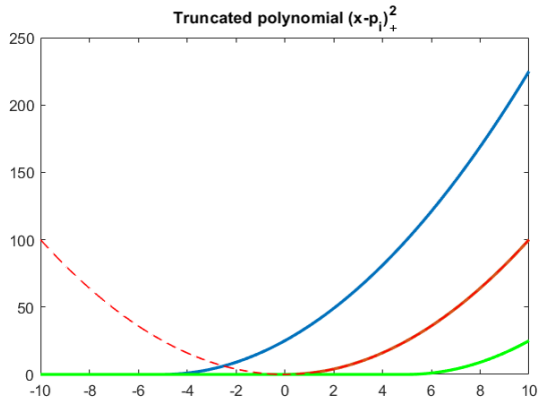
A **spline of degree** n is a function that, for given nodes $P = \{x_1, \ldots, x_N\} \subset \mathbb{R}$, coincides with polynomials of degree n on each open interval (x_i, x_{i+1}), (x_n, ∞), and $(-\infty, x_1)$ and whose derivatives up to order $n-1$ are continuous at all points $x_1, \ldots, x_N$.

We denote the space of functions comprising such splines by $\mathrm{SP}_{P,n}$.

Examples of elements of $\mathrm{SP}_{P,n}$ are the truncated polynomials

$$(x - p_i)_+^n = \begin{cases} (x-p_i)^n, & \text{if } x \geq p_i, \\ 0, & \text{else.} \end{cases}$$

Taking the n-th derivative $S^{(n)}$ of a spline S in $\mathrm{SP}_{P,n}$ leads to a step function with jumps at x_i.



Splines 1-D

The dimension of $\mathrm{SP}_{n,P}$ is

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

This can be seen from the following considerations: On each interval (x_i, x_{i+1}) we have $n+1$ degrees of freedom. Due to the differentiability condition at the nodes we loose n degrees of freedom at every node. Therefore, $\dim(\mathrm{SP}_{n,P}) = (N+1)(n+1) - Nn = n + N + 1$.

Splines 1-D

The dimension of $\mathrm{SP}_{n,P}$ is

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

This can be seen from the following considerations: On each interval (x_i, x_{i+1}) we have $n+1$ degrees of freedom. Due to the differentiability condition at the nodes we loose n degrees of freedom at every node. Therefore, $\dim(\mathrm{SP}_{n,P}) = (N+1)(n+1) - Nn = n + N + 1$.

A spline S can therefore be expressed, e.g., in the form

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

with weights $\lambda_1, \ldots, \lambda_{n+N+1}$ that need to be determined by a system of linear equations similar to polynomial interpolation.

Splines 1-D

The dimension of $\mathrm{SP}_{n,P}$ is

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

This can be seen from the following considerations: On each interval (x_i, x_{i+1}) we have $n+1$ degrees of freedom. Due to the differentiability condition at the nodes we loose n degrees of freedom at every node. Therefore, $\dim(\mathrm{SP}_{n,P}) = (N+1)(n+1) - Nn = n + N + 1$.

A spline S can therefore be expressed, e.g., in the form

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

with weights $\lambda_1, \ldots, \lambda_{n+N+1}$ that need to be determined by a system of linear equations similar to polynomial interpolation.

Typically, we only have one measurement y_i at every node x_i, i.e., N konstraints. In order to make the solution unique, we require $n+1$ additional constraints. In the case of odd $n = 2m - 1$, this can be

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

Splines that satisfy such conditions are called **natural splines**. We denote their space by $\mathrm{SPN}_{P,n}$.

Splines 1-D

Why are splines so nice? If values y_i are given in the points $x_1, \ldots, y_N \in \mathbb{R}$, then the spline $S \in \mathrm{SPN}_{3,P}$ interpolating

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

is the interpolating function that minimizes

$$\min_{f \text{ interpolates } P} \int_{x_1}^{x_N} |f''(x)|^2 dx.$$

These splines are are called natural **cubic splines**. They show some of the simplicity of polynomial interpolation but add stability and reduce oscillations.

Splines 1-D

Why are splines so nice? If values y_i are given in the points $x_1, \ldots, y_N \in \mathbb{R}$, then the spline $S \in \mathrm{SPN}_{3,P}$ interpolating

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

is the interpolating function that minimizes

$$\min_{f \text{ interpolates } P} \int_{x_1}^{x_N} |f''(x)|^2 dx.$$

These splines are are called natural **cubic splines**. They show some of the simplicity of polynomial interpolation but add stability and reduce oscillations. $\longrightarrow$ Matlab Example!

Properties:

- $f(x)$ is continuous and differentiable (n-times between data points, $(n-1)$-times at data points)
- global interpolator that relies on **all** (x_i, y_i) with $1 \leq i \leq N$
- extrapolation is technically possible

Spline n-D / Radial Basis Functions (RBFs)

If we are given a rectilinear grid of nodes, the 1-D splines can be generalized fairly easily to higher dimensions by tensor products. However, typically data is not given on rectilinear grids. Then we must rely on other methods. One possibility are radial basis functions.

If values y_i are given in the points $x_1, \ldots, x_N \in \mathbb{R}^2$, then the so-called **thin plate spline**

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

with $\Phi(r) = r^2 \ln(r)$, interpolating

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

is the interpolating function that minimizes

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

Spline n-D / Radial Basis Functions (RBFs)

Any function $\Phi : [0, \infty) \to \mathbb{R}$ which leads to a function $\phi(x) = \Phi(|x|)$, that only depends on the length of a vector, can be called a **radial basis function**. Typically, further 'nice' properties such as positive definiteness are desired. An approximating function based on this setup takes the form

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

with $\phi_i(x) = \Phi(|x - x_i|)$ and

$$S(x_i) = y_i, \quad i = 1, \ldots, N.$$

Spline n-D / Radial Basis Functions (RBFs)

Any function $\Phi : [0, \infty) \to \mathbb{R}$ which leads to a function $\phi(x) = \Phi(|x|)$, that only depends on the length of a vector, can be called a **radial basis function**. Typically, further 'nice' properties such as positive definiteness are desired. An approximating function based on this setup takes the form

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

with $\phi_i(x) = \Phi(|x - x_i|)$ and

$$S(x_i) = y_i, \quad i = 1, \ldots, N.$$

Popular choices for RBFs are

$$\Phi(r) = e^{-a^2 r^2}, \quad \text{(Gaussian)}$$
$$\Phi(r) = (r^2 + a^2)^{\frac{1}{2}}, \quad \text{(Multiquadric)}$$
$$\Phi(r) = \frac{1}{r^2 + a^2}, \quad \text{(Inverse Quadric)}$$
$$\Phi(r) = r^2 \ln(r), \quad \text{(Thin Plate Splines)}$$
compactly supported RBFs: more complicated

Spline n-D / Radial Basis Functions (RBFs)

Any function $\Phi : [0, \infty) \to \mathbb{R}$ which leads to a function $\phi(x) = \Phi(|x|)$, that only depends on the length of a vector, can be called a **radial basis function**. Typically, further 'nice' properties such as positive definiteness are desired. An approximating function based on this setup takes the form

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

with $\phi_i(x) = \Phi(|x - x_i|)$ and

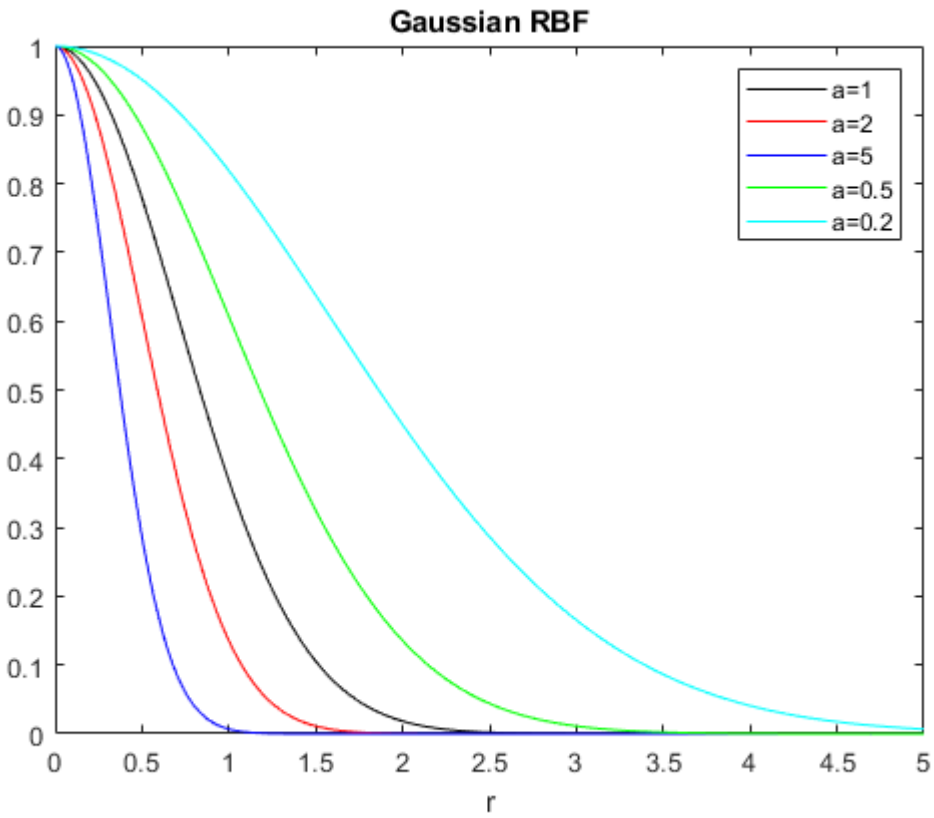
$$S(x_i) = y_i, \quad i = 1, \ldots, N.$$

Popular choices for RBFs are

$$\Phi(r) = e^{-a^2 r^2}, \quad \text{(Gaussian)}$$
$$\Phi(r) = (r^2 + a^2)^{\frac{1}{2}}, \quad \text{(Multiquadric)}$$
$$\Phi(r) = \frac{1}{r^2 + a^2}, \quad \text{(Inverse Quadric)}$$
$$\Phi(r) = r^2 \ln(r), \quad \text{(Thin Plate Splines)}$$
compactly supported RBFs: more complicated

Spline n-D / Radial Basis Functions (RBFs)

Any function $\Phi : [0, \infty) \to \mathbb{R}$ which leads to a function $\phi(x) = \Phi(|x|)$, that only depends on the length of a vector, such as positive definiteness

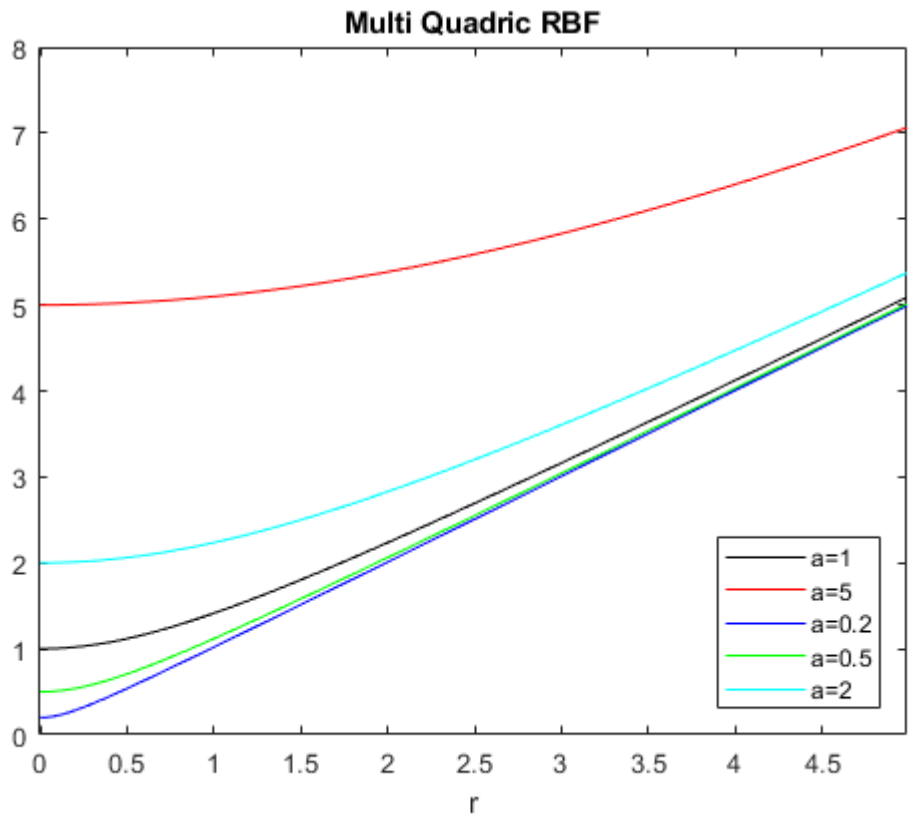


with $\phi_i(x) =$

Popular choi

Spline n-D / Radial Basis Functions (RBFs)

Any function $\Phi : [0, \infty) \to \mathbb{R}$ which leads to a function $\phi(x) = \Phi(|x|)$, that only depends on the length of a vector, as positive definiteness



with $\phi_i(x) =$

Popular choi

Spline n-D / Radial Basis Functions (RBFs)

Finding the coefficients $\lambda_0, \ldots, \lambda_m$ corresponds to solving the following system of linear equations:

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

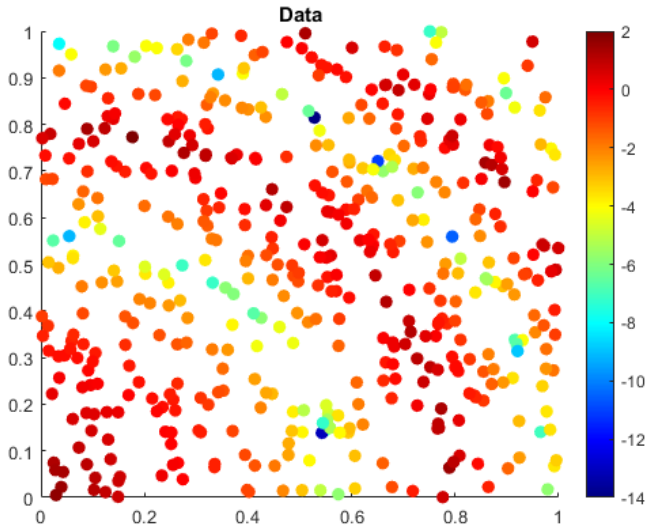
with $\boldsymbol{\lambda} = (\lambda_0, \ldots, \lambda_N)^T$, $\mathbf{y} = (y_1, \ldots, y_N)^T$ and

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

The unique solvability and the quality of the interpolation can strongly depend on the choice of the type of radial basis function.

Spline n-D / Radial Basis Functions (RBFs)

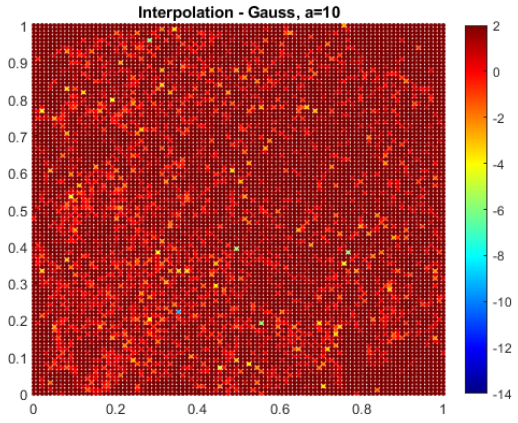
Finding the coefficients λ of the interpolation function s leads to the system of linear equations:

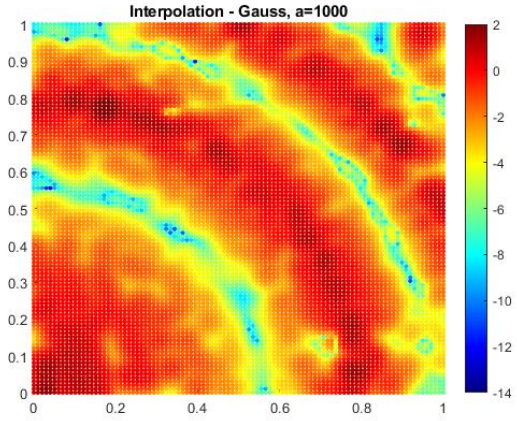


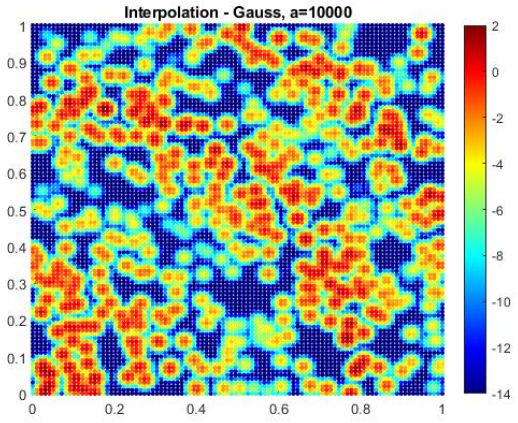
with $\boldsymbol{\lambda} = (\lambda_0, \ldots, \lambda_N)^T$,

$$\mathbf{M} \begin{pmatrix} \Phi(|x_1 - x_N|) \\ \Phi(|x_1 - x_2|) \\ \vdots \\ \Phi(0) \end{pmatrix}.$$

The unique solvability and the quality of the interpolation can strongly depend on the choice of the type







Spline n-D / Radial Basis Functions (RBFs)

Finding the coefficients $\lambda_0, \ldots, \lambda_m$ corresponds to solving the following system of linear equations:

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

with $\boldsymbol{\lambda} = (\lambda_0, \ldots, \lambda_N)^T$, $\mathbf{y} = (y_1, \ldots, y_N)^T$ and

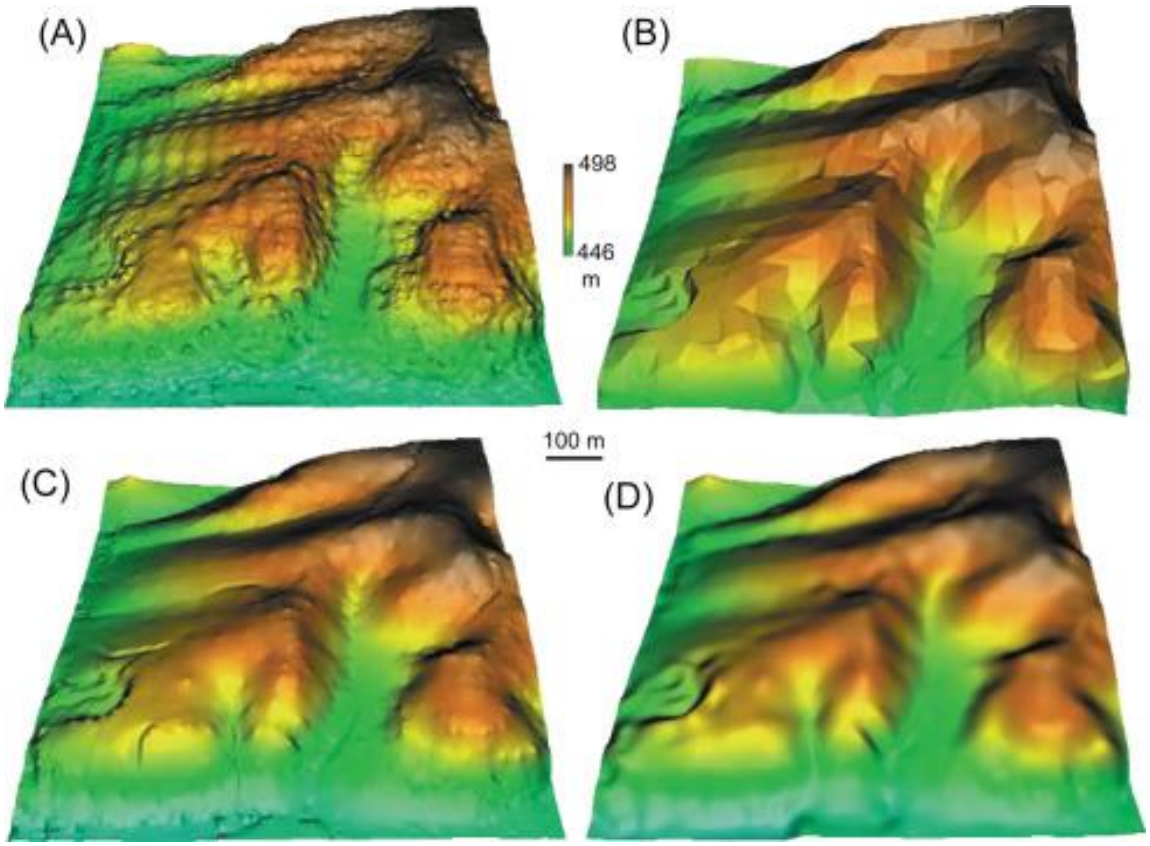
$$\mathbf{M} = \begin{pmatrix} \Phi(0) & \Phi(|x_1 - x_2|) & \Phi(|x_1 - x_3|) \cdots \Phi(|x_1 - x_N|) \\ \Phi(|x_2 - x_1|) & \Phi(0) & \Phi(|x_2 - x_3|) \cdots \Phi(|x_1 - x_2|) \\ \vdots & \vdots & \vdots \quad \vdots \quad \vdots \\ \Phi(|x_N - x_1|) & \Phi(|x_N - x_2|) & \Phi(|x_N - x_3|) \cdots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

The unique solvability and the quality of the interpolation can strongly depend on the choice of the type of radial basis function.

Properties:

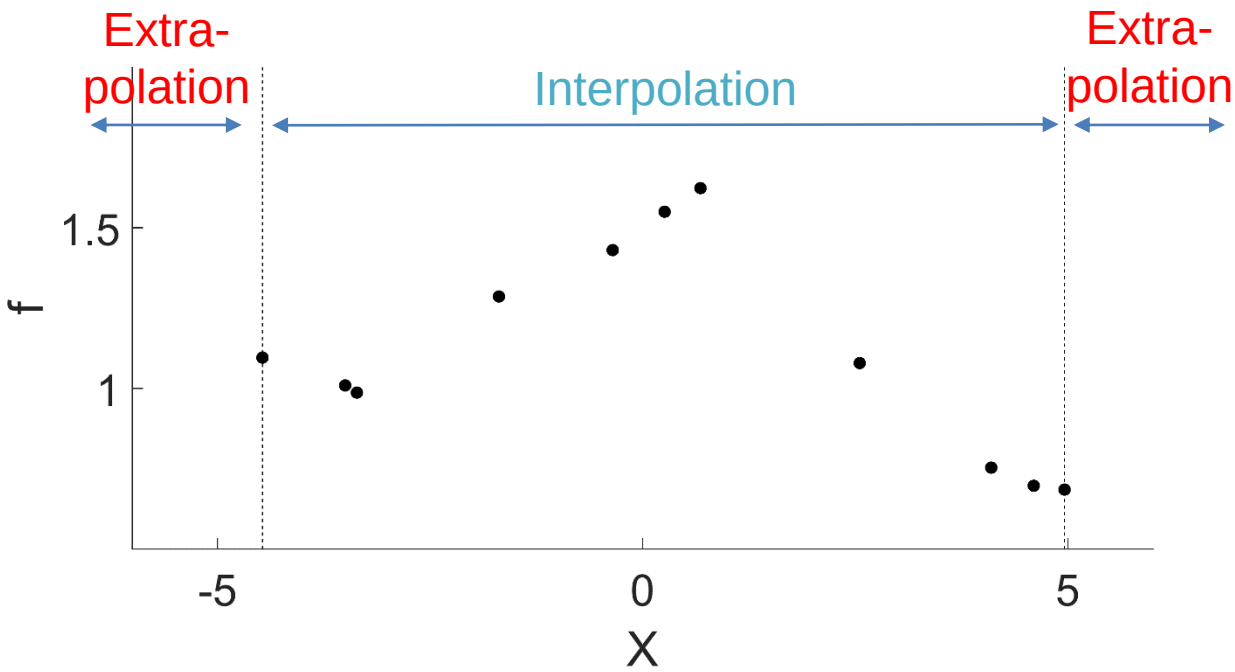
- $f(x)$ is continuous and differentiable
- global interpolator that relies on **all** (x_i, y_i) with $1 \leq i \leq N$
- extrapolation is technically possible

Example IDW, Lin. Interp. on Triangulation, Splines



Interpolation vs Extrapolation

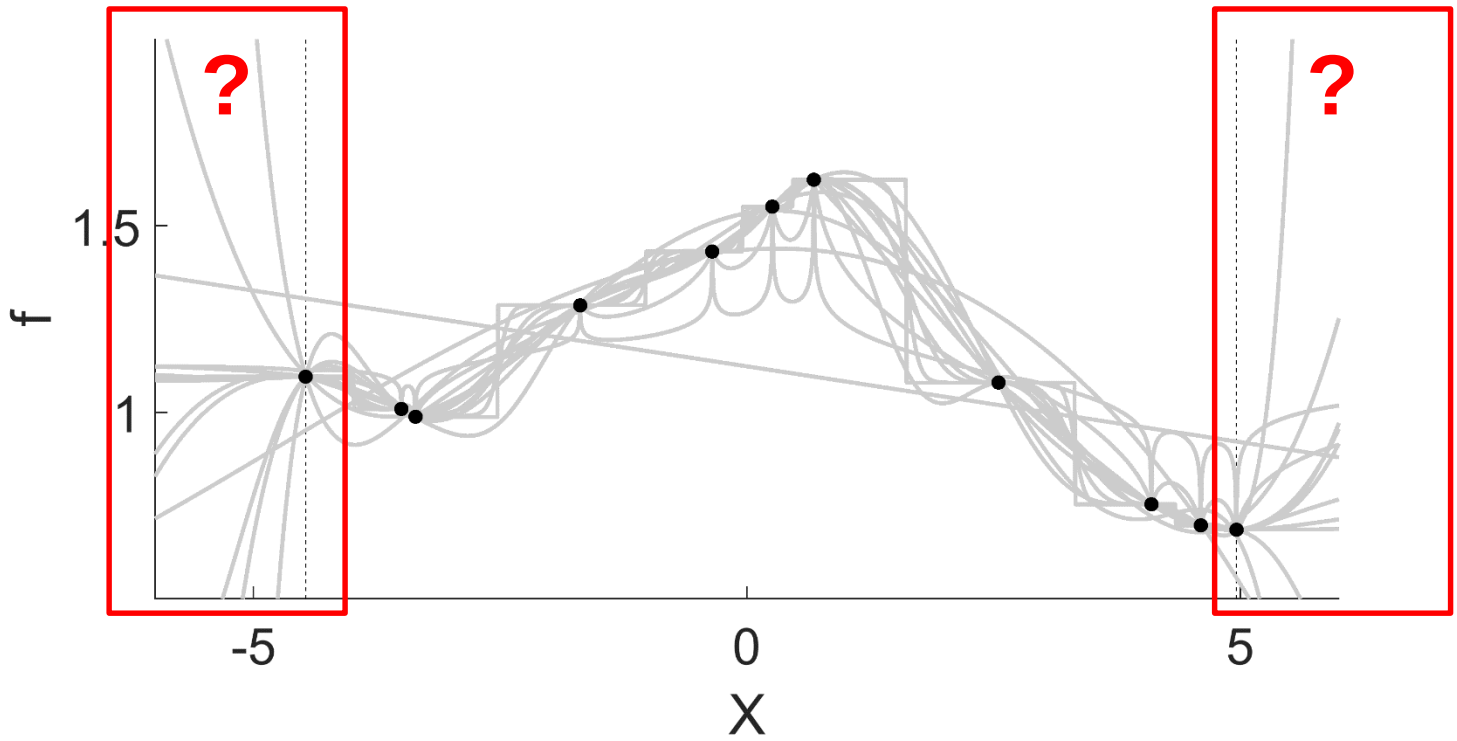
- **Inter**polation: Vorhersage **innerhalb** der Daten / **zwischen** den Datenpunkten
- **Extra**polation: Vorhersage **außerhalb** der Daten (zumeist außerhalb der konvexen Hülle)



Interpolation vs Extrapolation

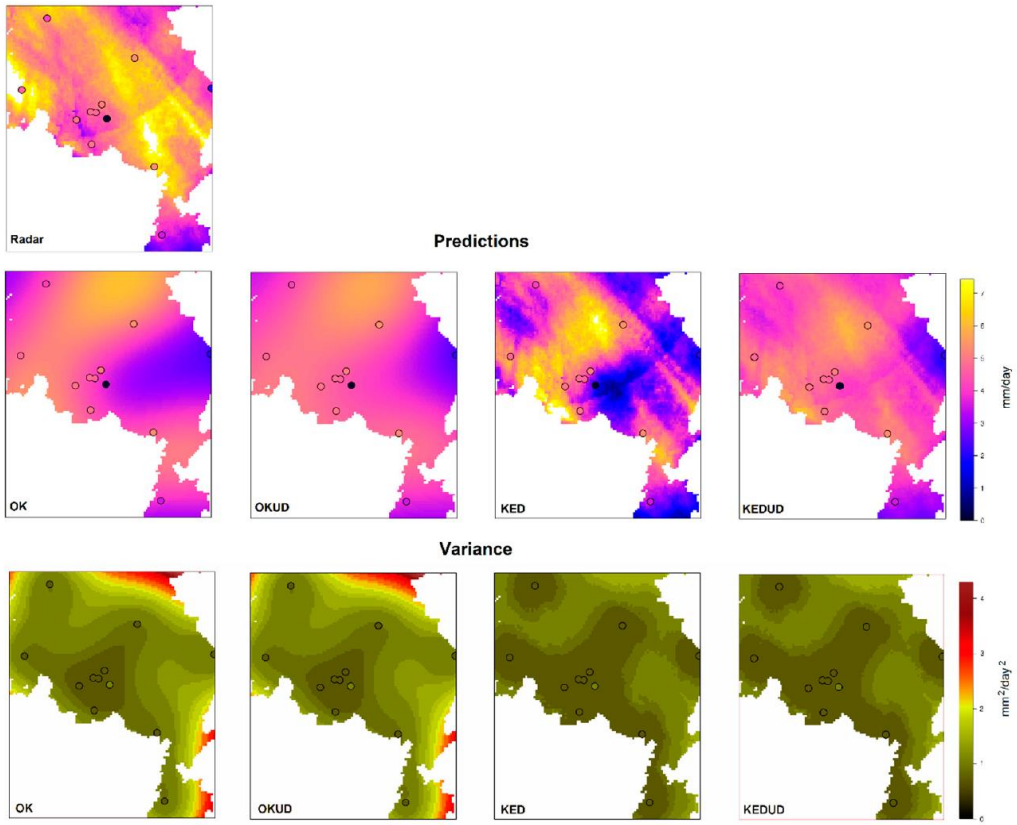
Was ist das Problem bei Extrapolation?

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



Statistical Methods

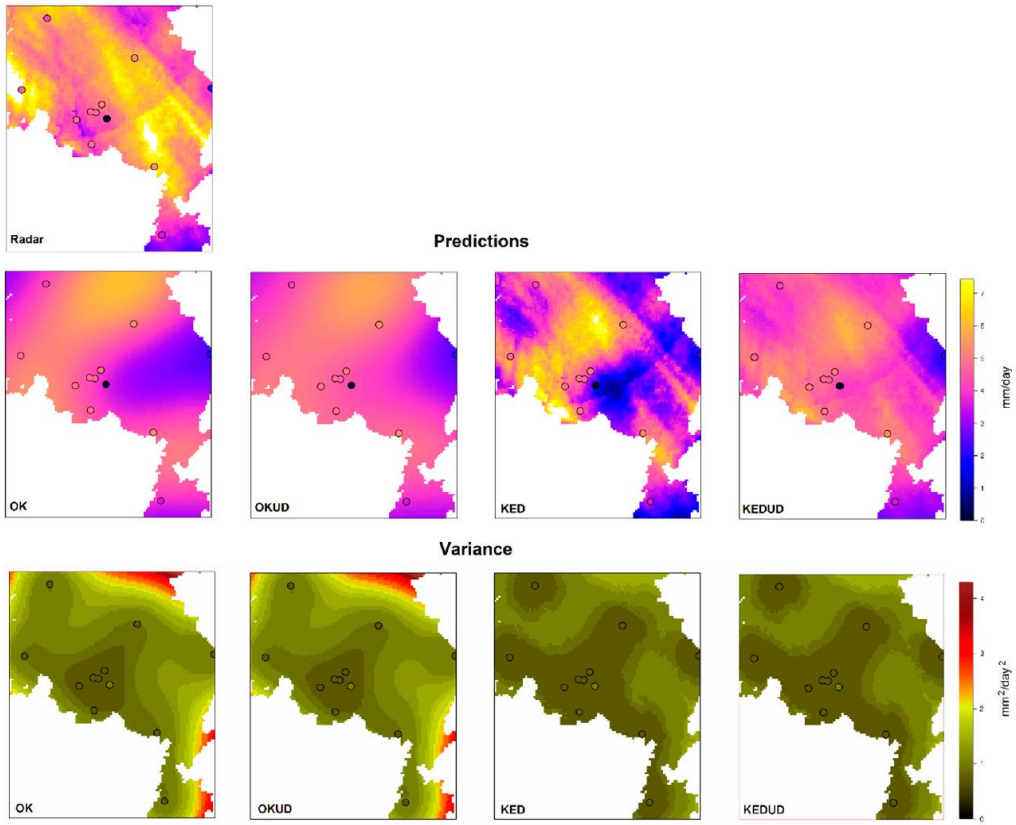
- **Kriging (Simple, Ordinary, Universal)**
- **Conditional Simulation**



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

The measurement Z_i at location $x_i \in \mathbb{R}^n$ is interpreted as a **random variable** with **expected value**

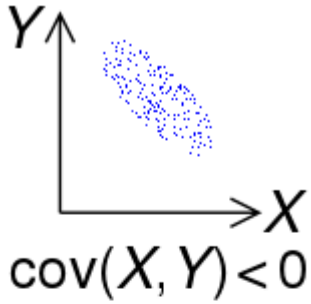
$$E[Z_i] = \int_\Omega z \, d\mu_{Z_i}(z)$$

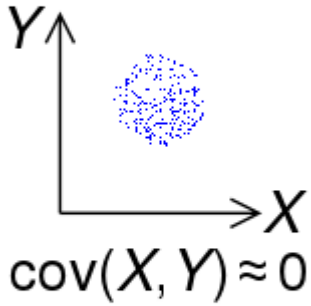
and **variance**

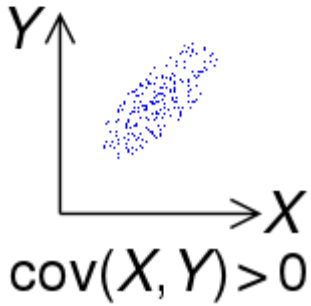
$$Var(Z_i) = E[(Z_i - E[Z_i])^2].$$

The **covariance** of two random variables is given by

$$Cov(Z_i, Z_j) = E[(Z_i - E[Z_i])(Z_j - E[Z_j])].$$







Kriging

The measurement Z_i at location $x_i \in \mathbb{R}^n$ is interpreted as a **random variable** with **expected value**

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

and **variance**

$$Var(Z_i) = E[(Z_i - E[Z_i])^2].$$

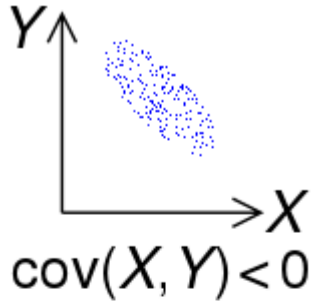
The **covariance** of two random variables is given by

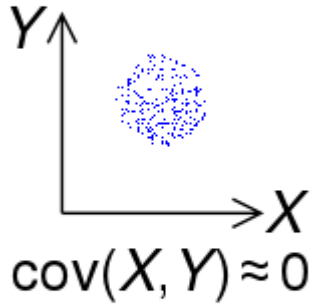
$$Cov(Z_i, Z_j) = E[(Z_i - E[Z_i])(Z_j - E[Z_j])].$$

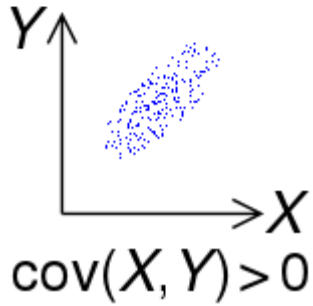
A crucial assumption in geostatistics is that of **stationarity**:*

$$E[Z(x)] = m, \quad Cov(Z(x+h), Z(x)) = C(h).$$

The covariance depends only on the increment $x - x_i$ between to locations, not the location itself!







* Stationarität oder auch Homogenität

Kriging

An approximation Z^* of Z is given by

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

with unknown weights $\lambda_i(x)$ such that

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

Kriging

An approximation Z^* of Z is given by

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

with unknown weights $\lambda_i(x)$ such that

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

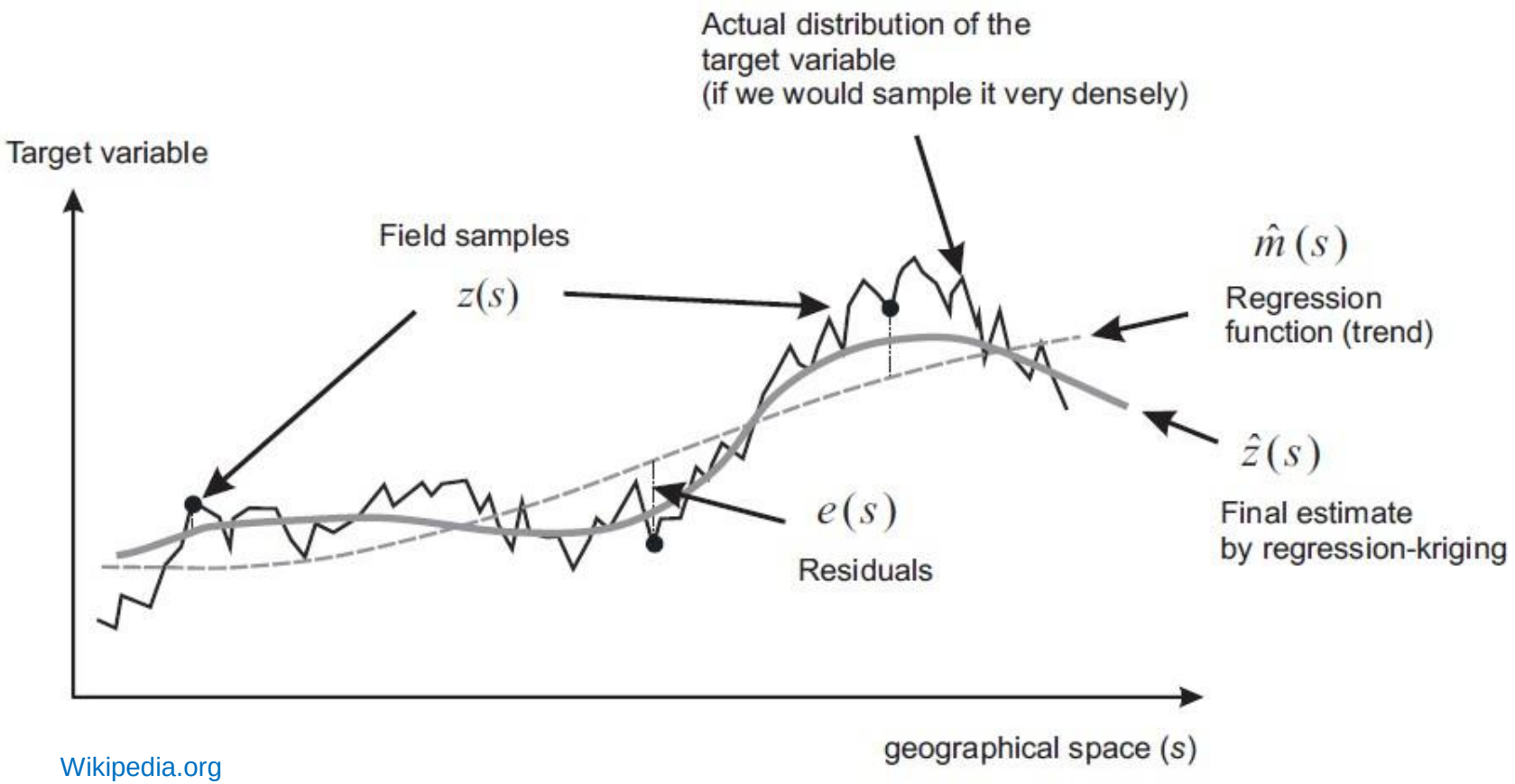
How to find the weights $\lambda_i(x)$? **Unbiasedness**

$$E[Z^*(x)] = E[Z(x)]$$

and **variance minimizing**

$$\min_{Z^*} Var(Z^*(x) - Z(x)).$$

Kriging



Wikipedia.org

Kriging

Simple Kriging: Expectation $m_0 = E[Z(x)]$ is known: Unbiasedness yields

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

Kriging

Simple Kriging: Expectation $m_0 = E[Z(x)]$ is known: Unbiasedness yields

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

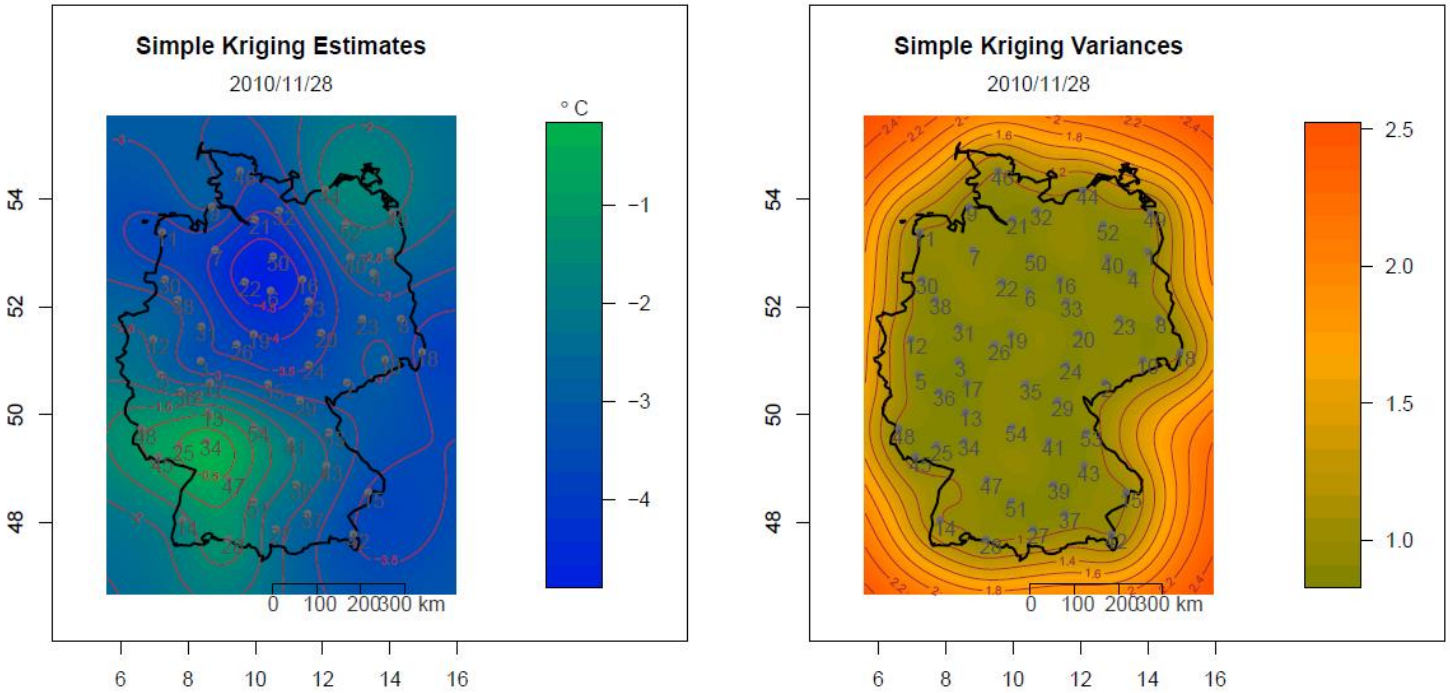
Variance minimization leads to

$$\sum_{i=1}^{n} \lambda_i(x) Cov(Z_i, Z_k) = Cov(Z_k, Z(x)), \quad k = 1, \ldots, n.$$

Note that stationarity implies $Cov(Z_k, Z(x)) = C(x_k - x) \longrightarrow$ Relation to radial basis functions.

Kriging

Simple Kriging: Expectation $m_0 = E[Z(x)]$ is known: Unbiasedness yields



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

Kriging

In general, $Cov(Z(x_i), Z(x_j))$ is unkown and needs to be estimated or modelled.

A crucial role is given to the **variogramm**

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

It can be estimated from the data (under certain conditions).

Kriging

In general, $Cov(Z(x_i), Z(x_j))$ is unkown and needs to be estimated or modelled.

A crucial role is given to the **variogramm**

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

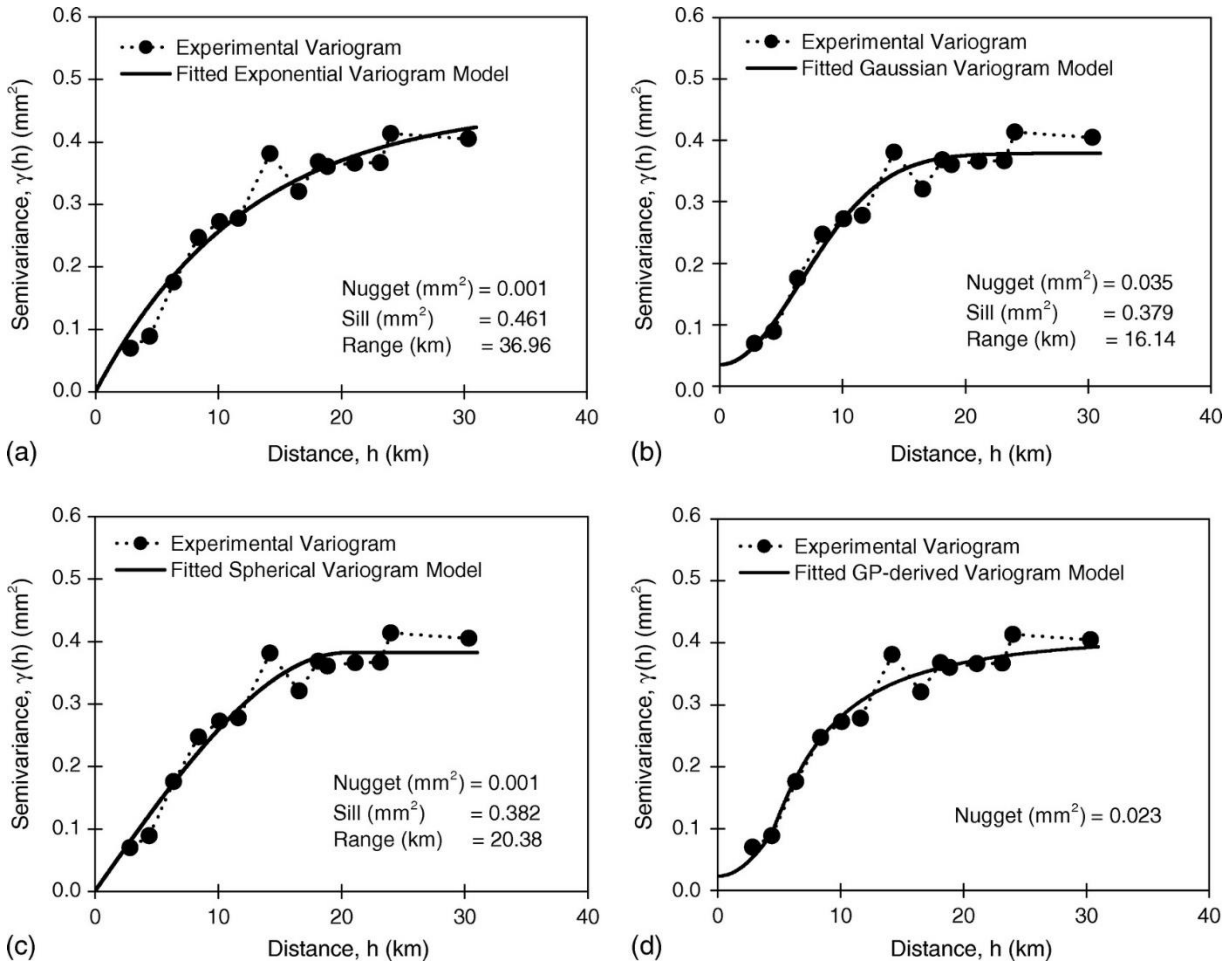
It can be estimated from the data (under certain conditions).

Empirical variogram

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

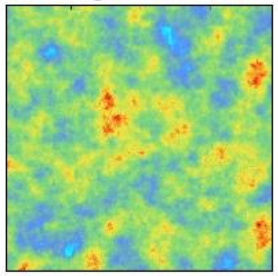
After the empirical variogram has been computed, a theoretical model needs to be fit to it. This model needs to satisfy certain properties, e.g. conditional negative definiteness.

Kriging

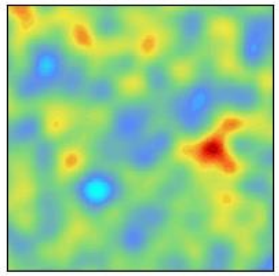

Experimental Variogram
Fitted Exponential Variogram Model
Nugget (mm²) = 0.001
Sill (mm²) = 0.461
Range (km) = 36.96
Semivariance, γ(h) (mm²)
Distance, h (km)
(a)
Experimental Variogram
Fitted Gaussian Variogram Model
Nugget (mm²) = 0.035
Sill (mm²) = 0.379
Range (km) = 16.14
Semivariance, γ(h) (mm²)
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

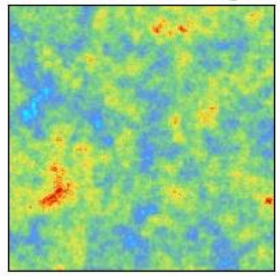
Rough



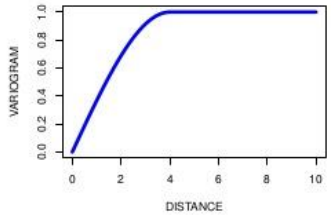
Smooth



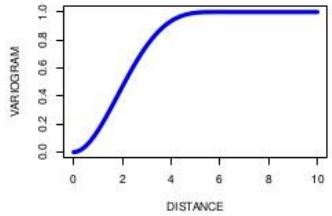
Rough



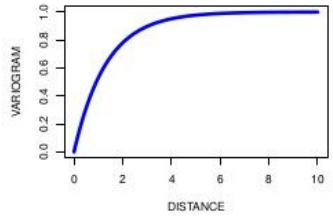
Spherical model



Cubic model

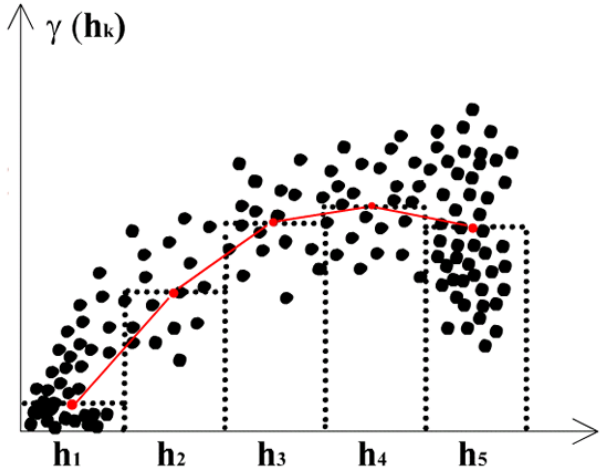


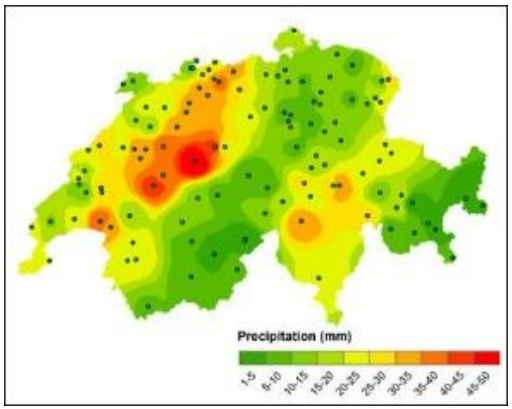
Exponential model



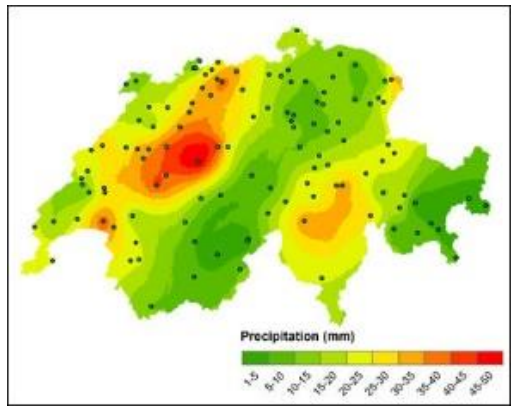
From a presentation by H. Wackernagel

Example IDW, Kriging





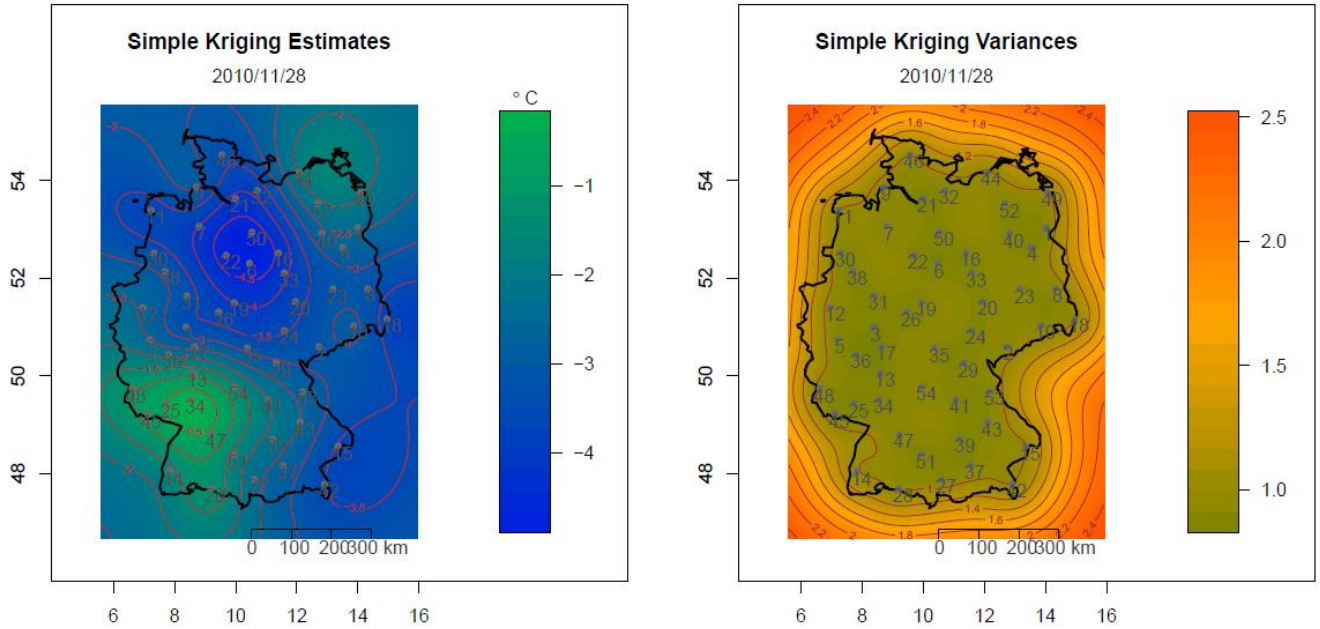
IDW



Kriging

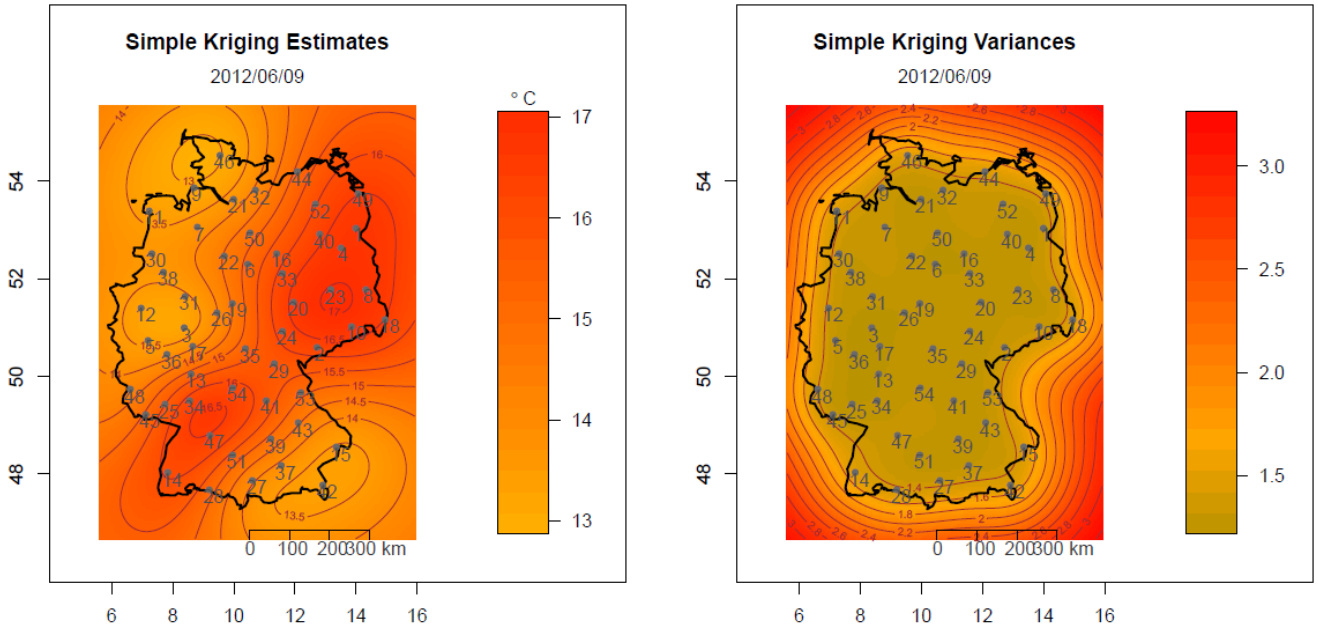
gitta.info

Example Kriging



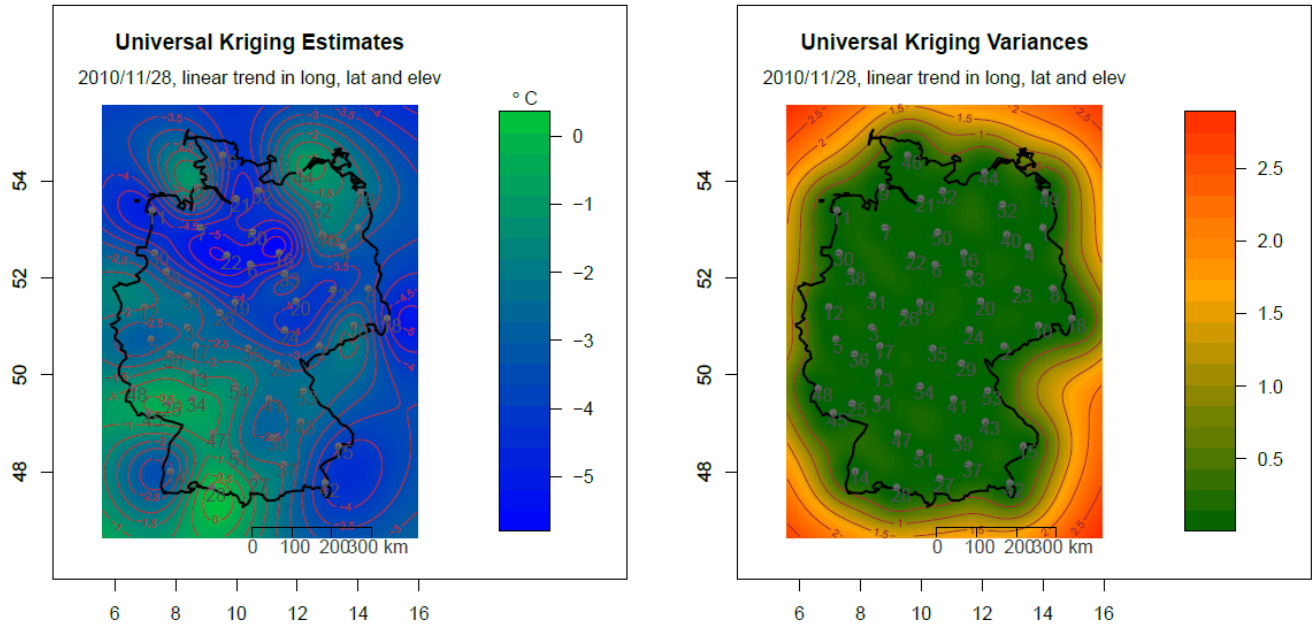
Example from the Bachelor thesis ”Kriging methods in spatial statistics”, Andreas Lichtenstern, TU München

Example Kriging



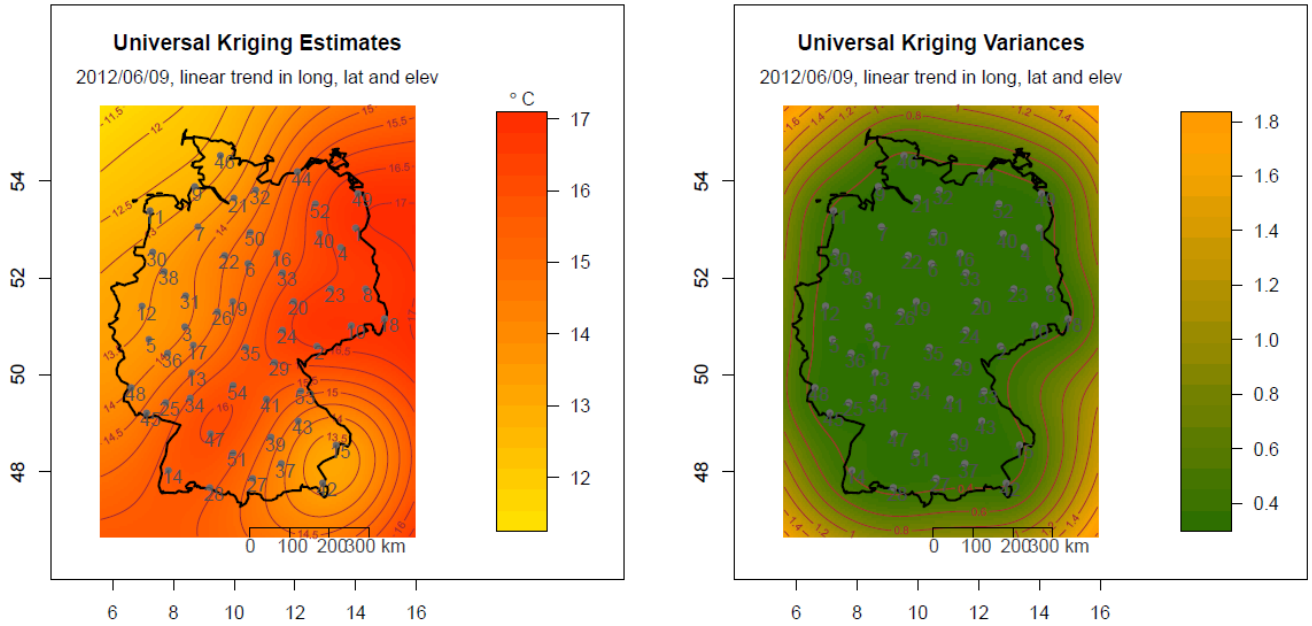
Example from the Bachelor thesis ”Kriging methods in spatial statistics”, Andreas Lichtenstern, TU München

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 (Vorlesung 1)
2. Koordinatensysteme und -transformationen (Vorlesung 2+3)
3. Räumliche Datenmodellierung (Vorlesung 4 - 6)
4. Vermaschungen (Vorlesung 7 - 8)
5. Räumliche Interpolation (8 - 10)
6. **Transformationen, Filtermethoden**

Filter Methods

Instead of the true image, we typically only know a **noisy image**

$$f = f_0 + n,$$

where the noise n can be of different type (typically Gaussian). **Denoising** describes the problem of obtaining a $g \approx f_0$ from knowledge of f.



Original Image



Image with Gaussian noise

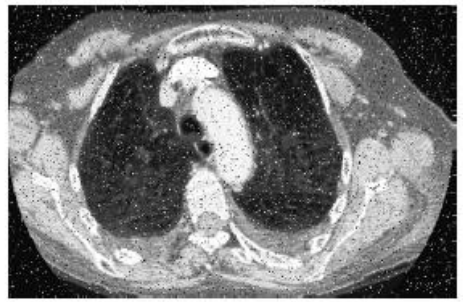


Image with Salt and Pepper noise

Filter Methods

Instead of the true image, we typically only know a **noisy image**

$$f = f_0 + n,$$

where the noise n can be of different type (typically Gaussian). **Denoising** describes the problem of obtaining a $g \approx f_0$ from knowledge of f.

Denoising is often an initial step for further image processing.



Original Image



Image with Gaussian noise

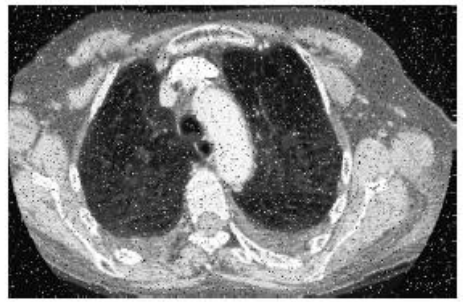


Image with Salt and Pepper noise

Filter Methods

Convolution: Given a kernel $w = [0,1]^2 \to \mathbb{R}$, the convolution of f agains w is defined via

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

Fourier Transform: The Fourier transform of f defined via

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

Convolution and Fourier Transform: The Fourier transform of a convolution has the property

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

Filter Methods

Convolution: Given a kernel $w = [0,1]^2 \to \mathbb{R}$, the convolution of f agains w is defined via

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

Fourier Transform: The Fourier transform of f defined via

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

Convolution and Fourier Transform: The Fourier transform of a convolution has the property

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

The choice of the kernel w or W in spatial or spectral domain, respectively, determines the properties of the convolution (filter).

Filter Methods

Representation in a discrete (pixel) setup:

Convolution: Given a kernel $w \in \mathbb{R}^{(2K+1)\times(2L+1)}$, the convolution of f agains w is defined via

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

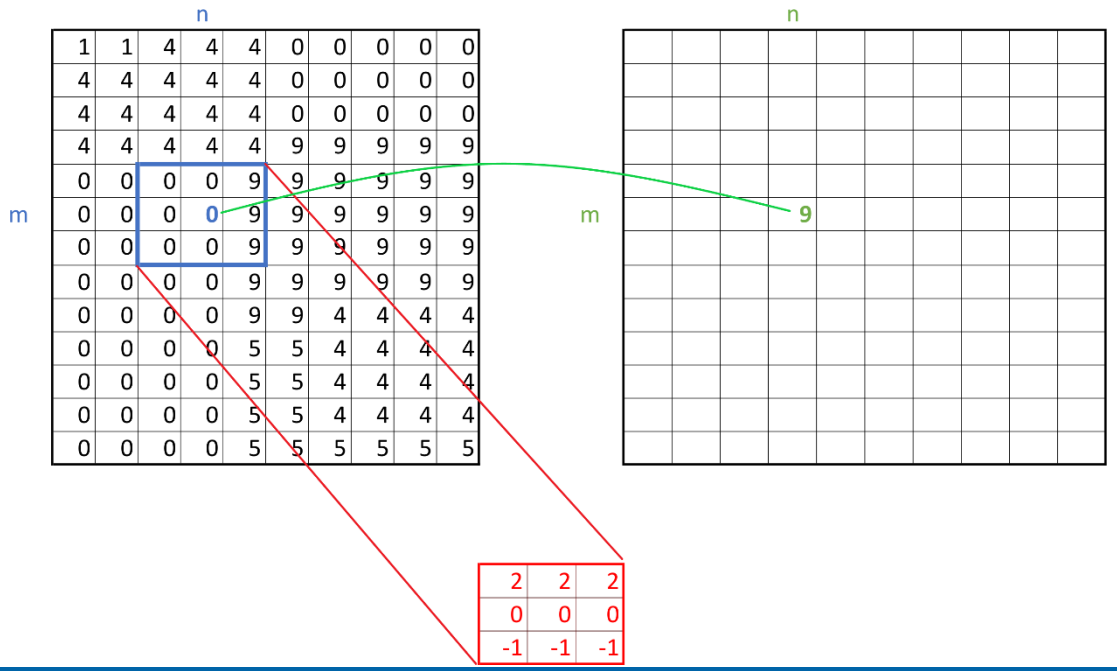
1	1	4	4	4	0	0	0	0	0
4	4	4	4	4	0	0	0	0	0
4	4	4	4	4	0	0	0	0	0
4	4	4	4	4	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	9	9	9	9
0	0	0	0	9	9	4	4	4	4
0	0	0	0	5	5	4	4	4	4
0	0	0	0	5	5	4	4	4	4
0	0	0	0	5	5	4	4	4	4
0	0	0	0	5	5	5	5	5	5

Filter Methods

Representation in a discrete (pixel) setup:

Convolution: Given a kernel $w \in \mathbb{R}^{(2K+1)\times(2L+1)}$, the convolution of f agains w is defined via

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



Filter Methods

Examples for kernels w:

Averaging Kernel in a $(2K+1) \times (2L+1)$-neighbourhood. For $L = K = 1$:

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

Gaussian Kernel with weight function $w(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{\sigma^2}}$. For $L = K = 2$ and $\sigma = 1$:

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

Gaussian Filtering can be regarded as a special case of diffusion filtering.

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 Methods

Original Image



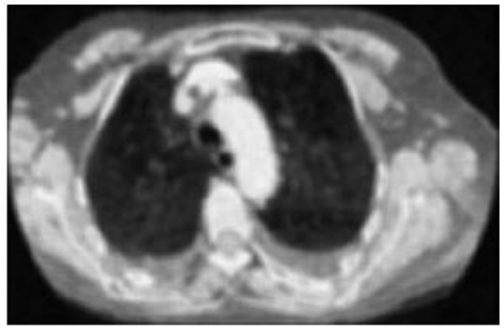
Image with Gaussian noise



Gauss Filtered Image



Average Filtered Image



Filter Methods

Examples for nonlinear methods:

Median Filter: For each pixel (m, n) we define a local environment $LA(m, n) = \{m - K, \ldots, m + K\} \times \{n - L, \ldots, n + L\}$ and choose the median intensity in that environment:

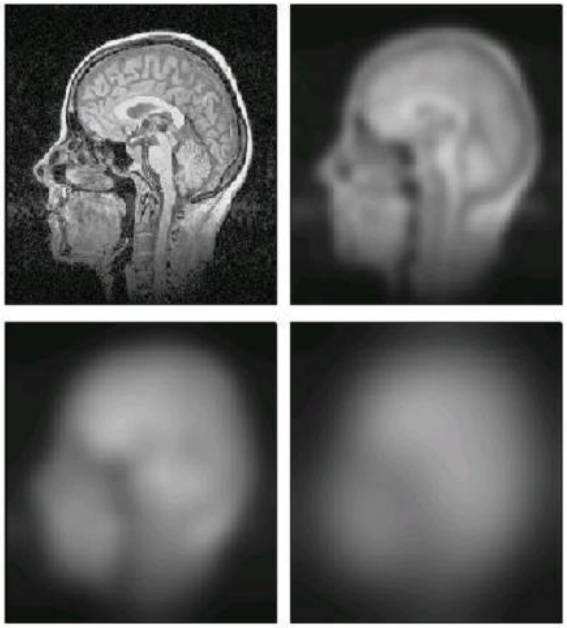
$$g(m, n) = \text{median}\{f(k, l) : (k, l) \in LA(m, n)\}$$

Nonlinear Diffusion: Let D denote the diffusion coefficient, which may depend on g and ∇g (latter accounts for 'edges'), then the diffusion equation reads

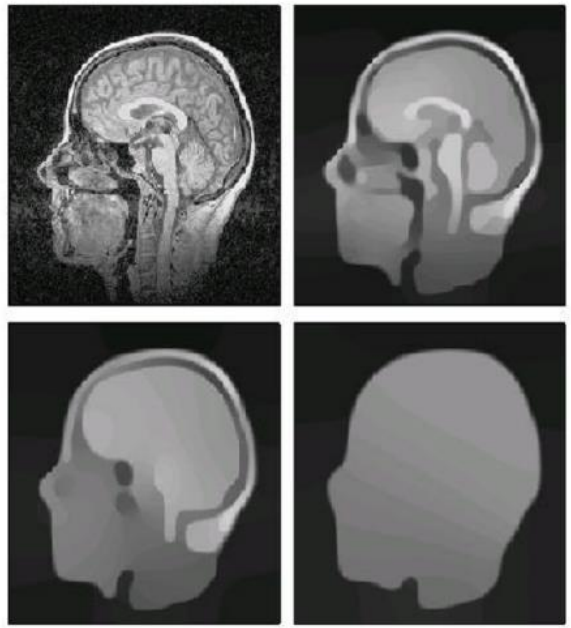
$$\begin{aligned} \frac{d}{dt} g(t, x, y) &= \nabla \cdot \big(D(g, \nabla g)\big) \nabla g(t, x, y) \\ g(0, x, y) &= f(x, y). \end{aligned}$$

A popular choice for D is $D(|\nabla g|) = e^{-\frac{|\nabla g|^2}{2\sigma^2}}$ (Perona-Malik diffusion)

Filter Methods



Scale space behaviour of **linear** Diffusion filtering



Scale space behaviour of **Nonlinear** Diffusion filtering

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

Filter Methods

Where else could nonlinear diffusion filtering be used?

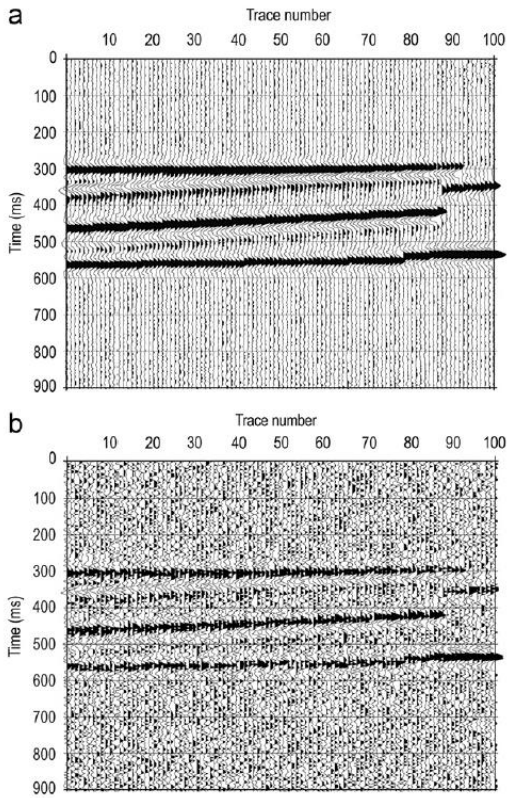


Fig. 3. Synthetic section contaminated with 10% (a) and 50% (b) random noise.

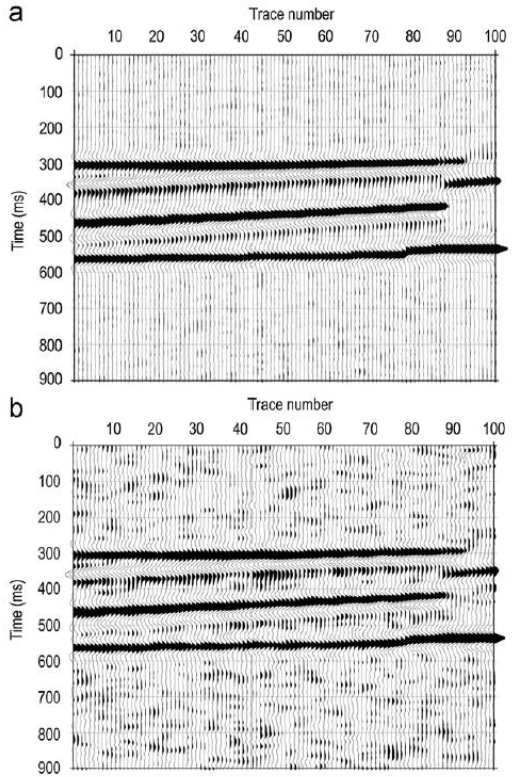
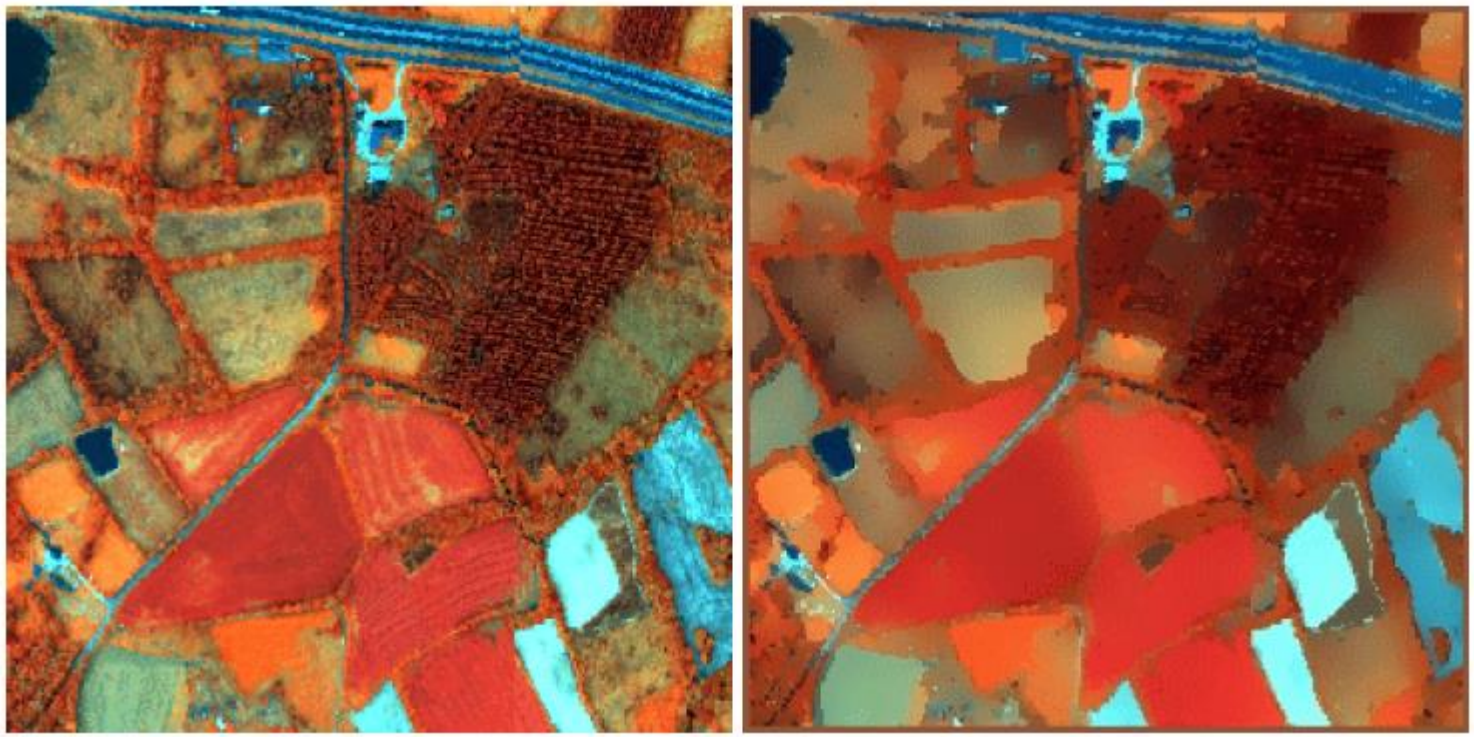


Fig. 4. Anisotropic non-linear diffusion filtering results: (a) section contaminated with 10% random noise and (b) section contaminated with 50% noise.

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

Filter Methods

Where else could nonlinear diffusion filtering be used?



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

Image Segmentation

Image segmentation refers to a wide range of applications. It can mean the separation of an image into foreground and background, the subdivision of an image into regions of similar properteis (e.g., clustering), or the subdivision into regions separated by certain structures.

The two main approaches to segmentation are therefore:

(1) **Discontinuities**: Separate image based on edges

(2) **Similarity**: Separate image based on similar properties

All in all, there is no universal segmentation technique that works for all problems. Manual segmentation is feasible only for small data sets.

Edge Detection

The visualization of edges in pictures is typically based on gradients. Widely used filters are:

Prewitt Filter: The following filter matrices for differentiation in x- and y-direction are chosen

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

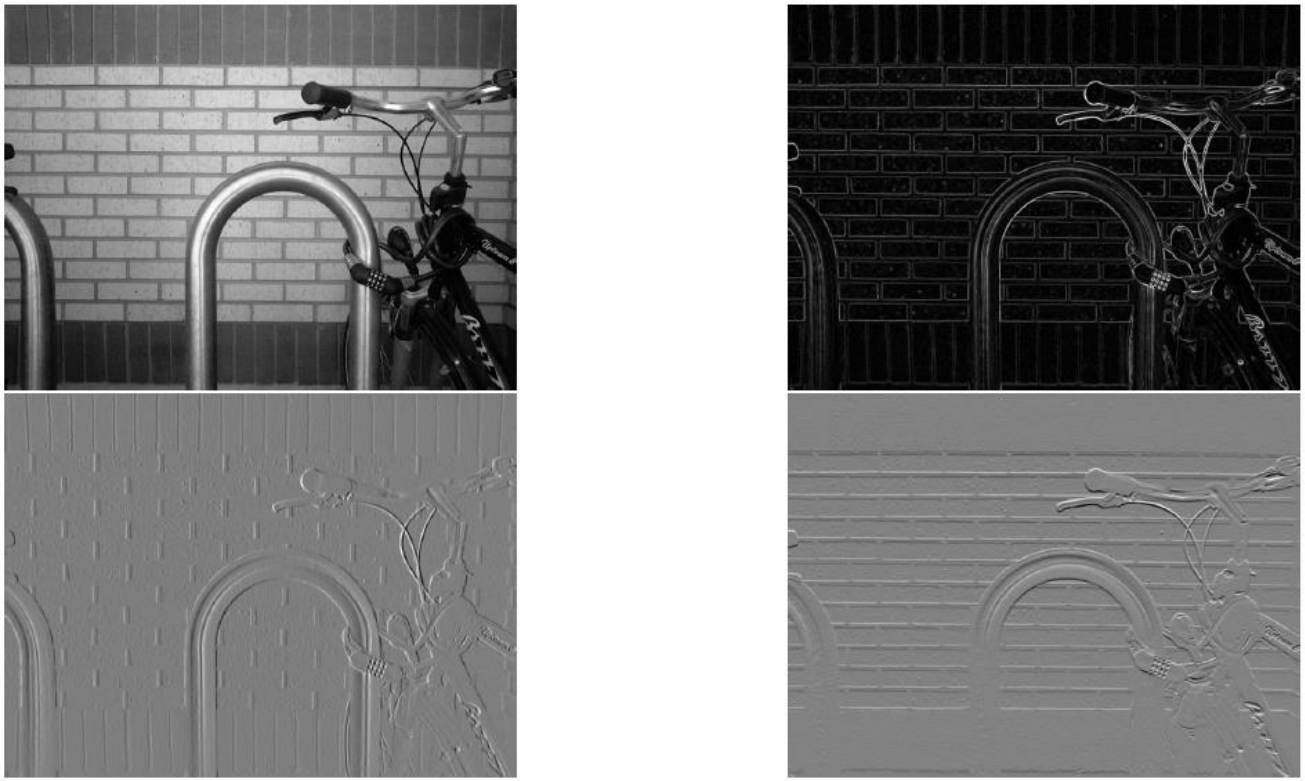
Sobel Filter: The following filter matrices for differentiation in x- and y-direction are chosen

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

Edges are those pixels, where the gradient is large. Alternatively, one can work with second derivatives: The second derivative is zero where the gradient is largest (Exercise: what could a filter matrix look like to determine the second derivative). A method working with second derivatives is, e.g., Laplace zero crossing.

Edge Detection

Top left: Original image, bottom left: Sobel filter in x direction, bottom right: Sobel filter in y-direction, top right: Sobel filter in x- and y- direction



Thresholding

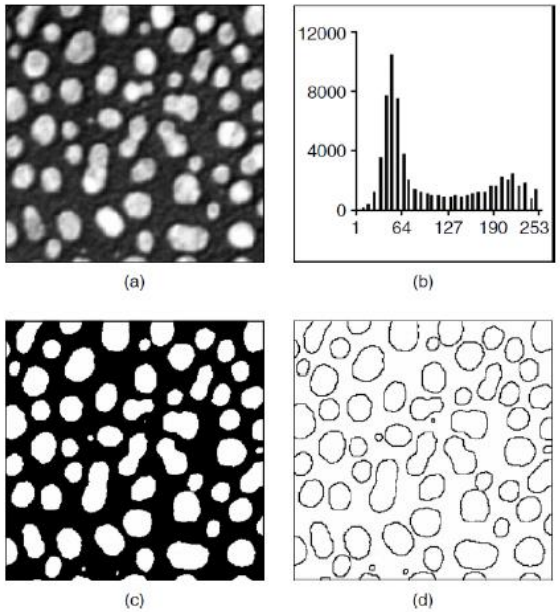
Thresholding is a technique that is based on similarity, namely a similar grayscale. It works well for images with **bimodal histograms**. Choose an **intensity threshold** $T > 0$ and define the segmented image g via

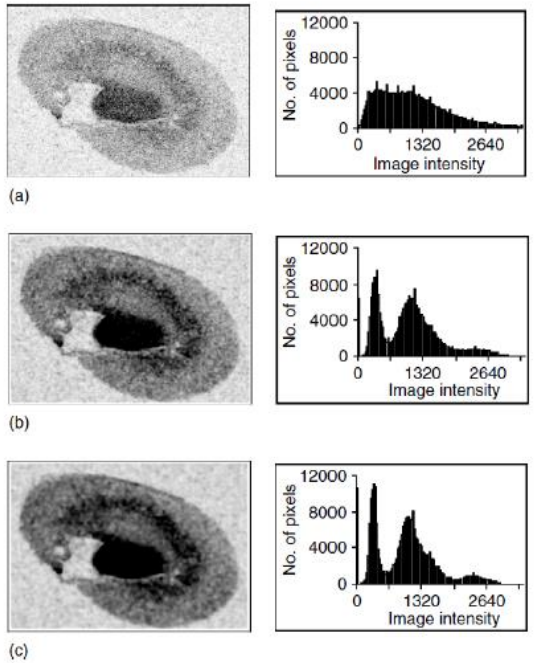
$$g(n,m) = \begin{cases} 1, & \text{if } f(n,m) > T, \\ 0, & \text{if } f(n,m) \leq T. \end{cases}$$

The areas where $g(n,m) = 1$ could be denoted as foreground and the rest as background.

Thresholding

Different segementation techniques can be combined. The **left bloc**k shows the following: (a) Original image, (b) its histogram, (c) after thresholding (d) after Laplace zero crossing. The **right block** shows: (a) original image and its histogram, (b)+(c) image and histogram after median filtering. The latter shows that filtering can turn images into bimodal images that can be segmented by thresholding.





Thresholding

When using thresholding, it is not always clear which threshold T to choose. An automatic method is, e.g., **Otsu's method**:

1.)Choose initial threshold $T = 0$

2.)Separate f into foreground g_1 and background g_0

3.)Compute the intra-class variance

$$\sigma(T)^2 = w_0(T)\sigma_0(T)^2 + w_1(T)\sigma_1(T)^2,$$

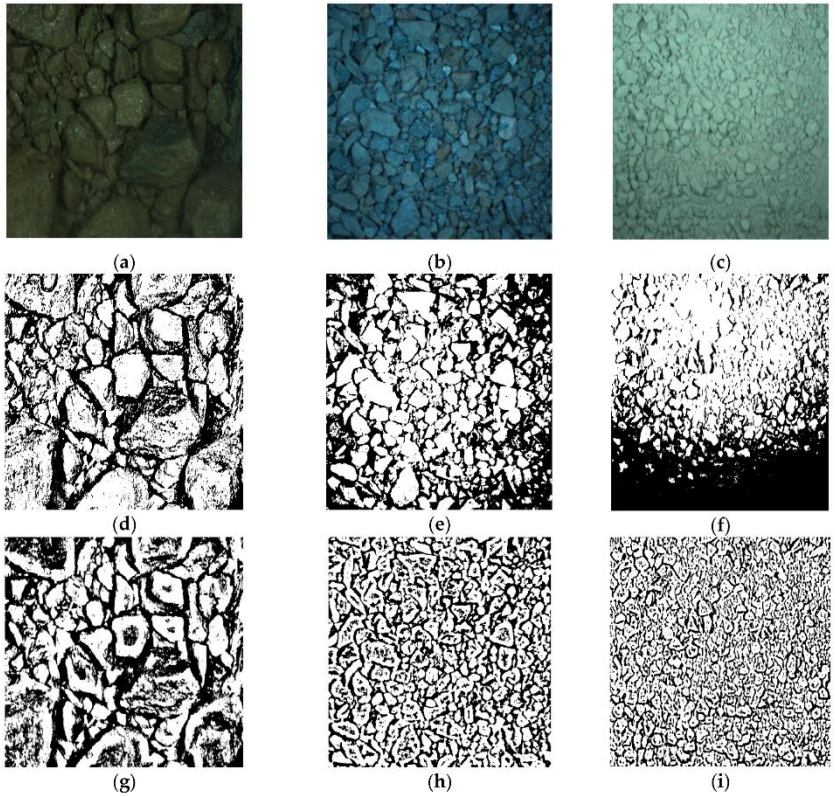
where w_0 is the relative number of pixels in the background, w_1 the relative number of pixels in the foreground, $\sigma_0(T)^2$ the variance in the background and $\sigma_1(T)^2$ the variance in the foreground

4.)Iterate for all possible thresholds T

5.)Choose the threshold with the smallest intraclass variance

Thresholding

Otsu's Method demonstrated for the segmentation of ore structures: Top: Original images of coarse, medium and fine ores, Center: after application of Otsu's method, Bottom: after application of an improved Otsu's method



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

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

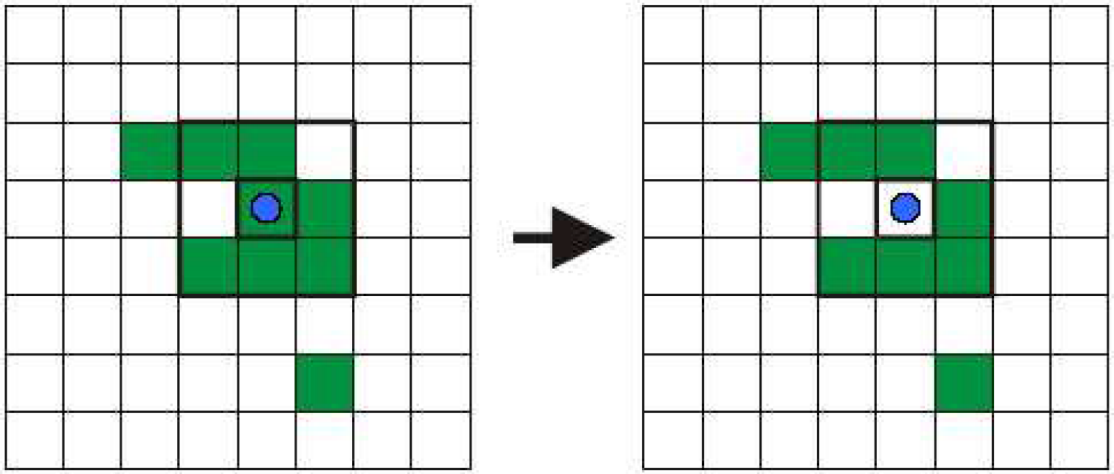
$$\check{X} = \{-p : p \in X\}$$

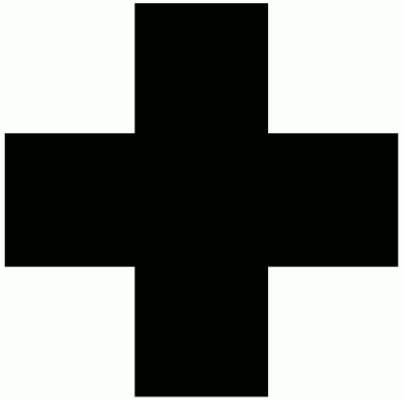
Morphologische Operationen

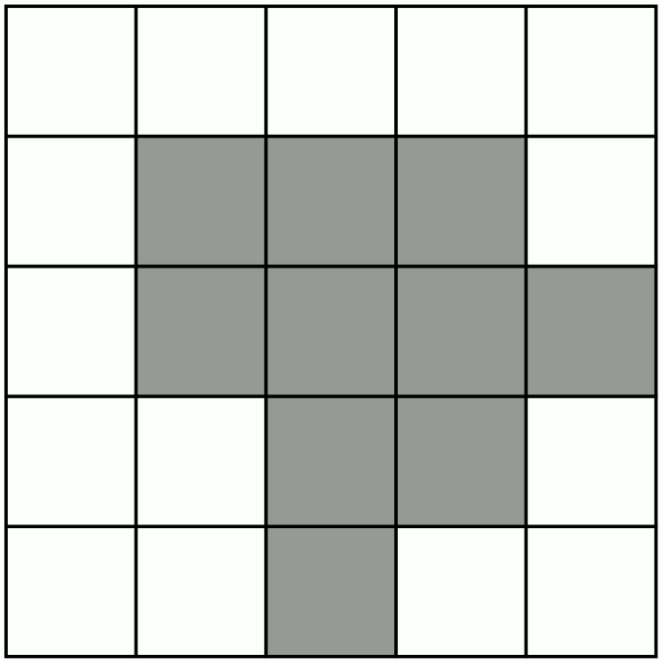
Die **Erosion** einer Menge X mit einem Strukturelement B ist nun definiert als

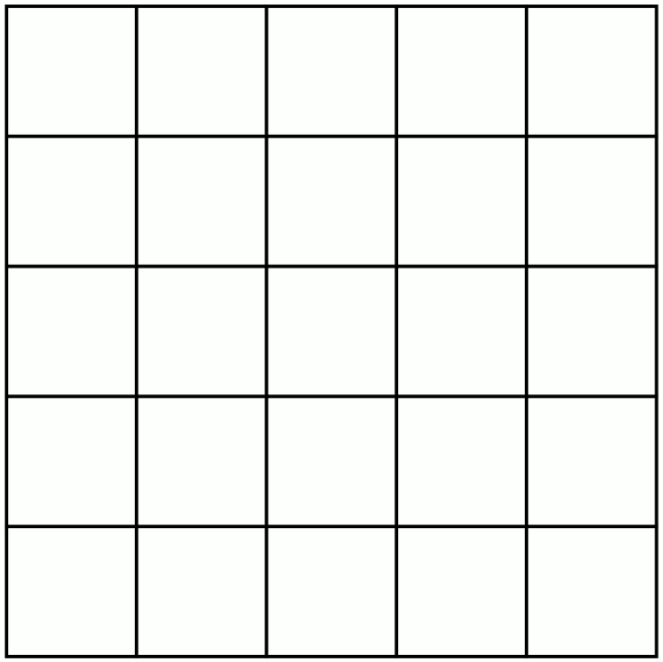
$$\varepsilon_B(X) = \{p \in \{1, \ldots, N\} \times \{1, \ldots, M\} : B_p \subset X\} = \bigcap_{p \in B} X_{-p}.$$

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









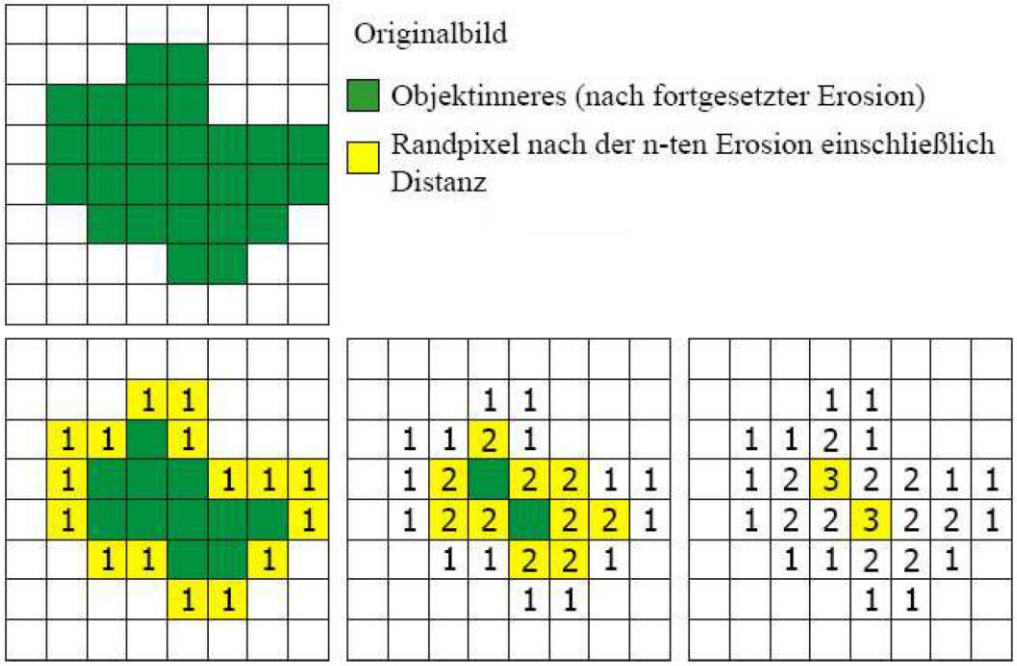
[Animation siehe OPAL!](https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975) (https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975)

Morphologische Operationen

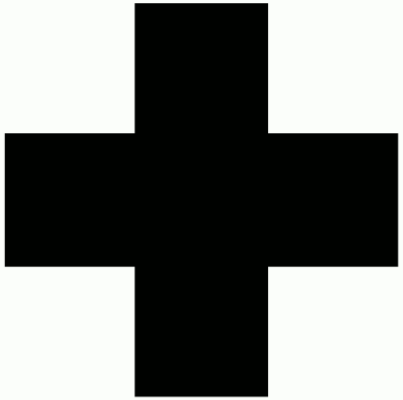
Die **Erosion** einer Menge X mit einem Strukturelement B ist nun definiert als

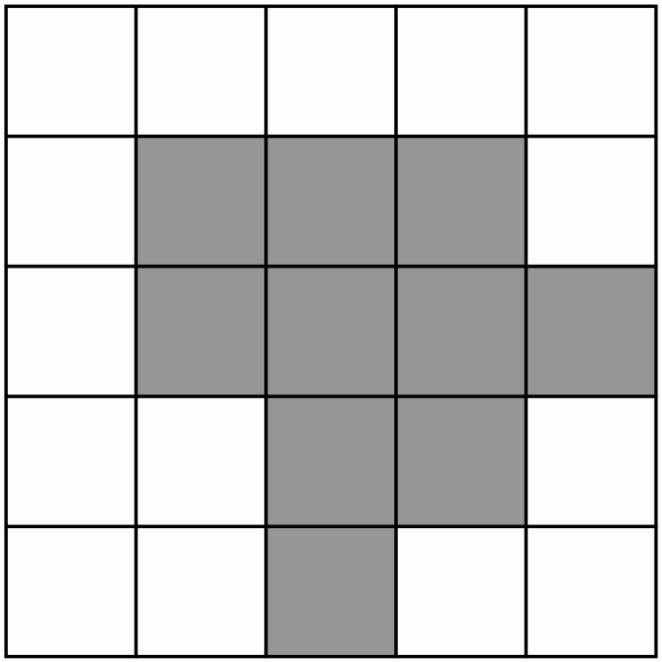
$$\varepsilon_B(X) = \{p \in \{1,\dots,N\} \times \{1,\dots,M\} : B_p \subset X\} = \bigcap_{p \in B} X_{-p}.$$

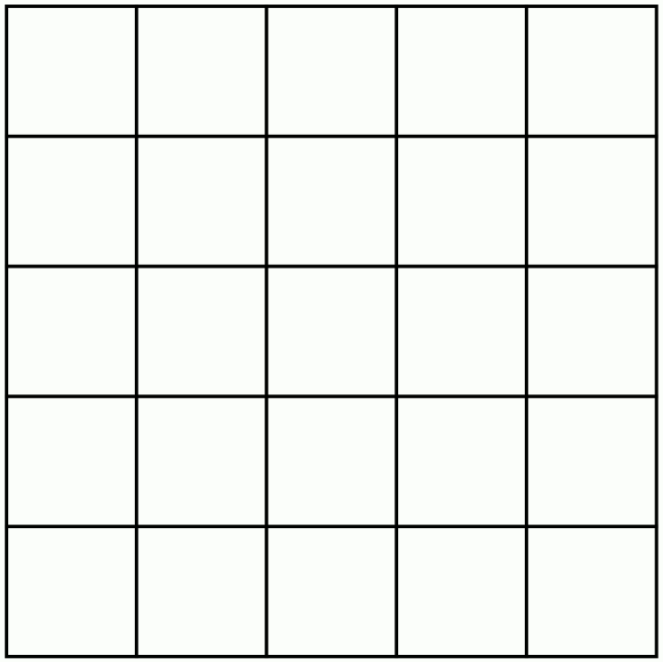
In Worten: Erosion beschreibt alle Pixel p, für die das Strukturelement B mit Zentrum p in X liegt.



Welches Strukturelement wurde hier gewählt?







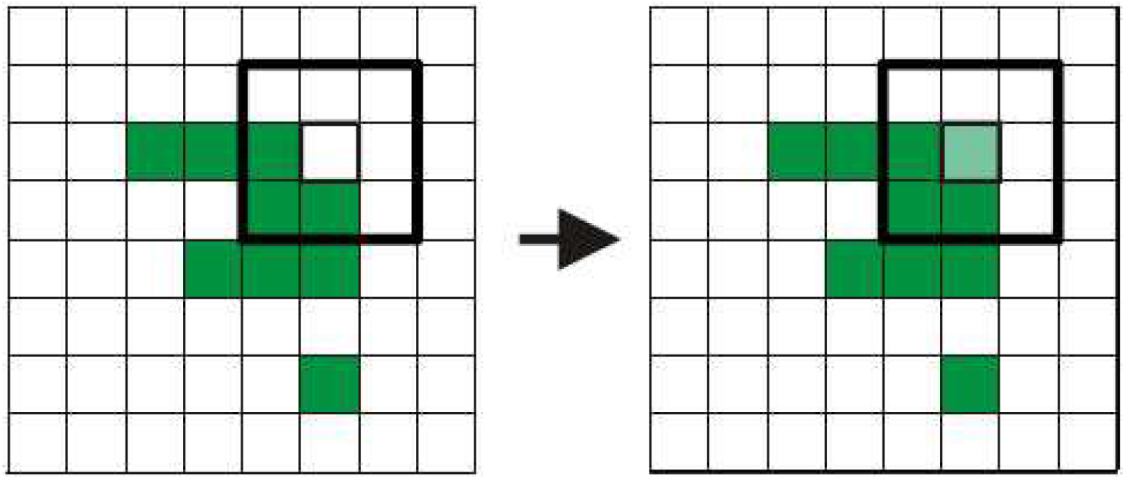
[Animation siehe OPAL!](https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975) (https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/19758415873/CourseNode/102027240864975)

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.



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

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

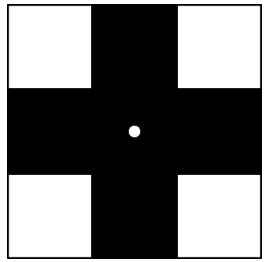
In Worten: öffnung beschreibt beschreibt die Vereinigung aller Strukturelemente B_p, die in X enhalten sind, nicht nur der Referenzpunkte p (lezteres ist Erosion).

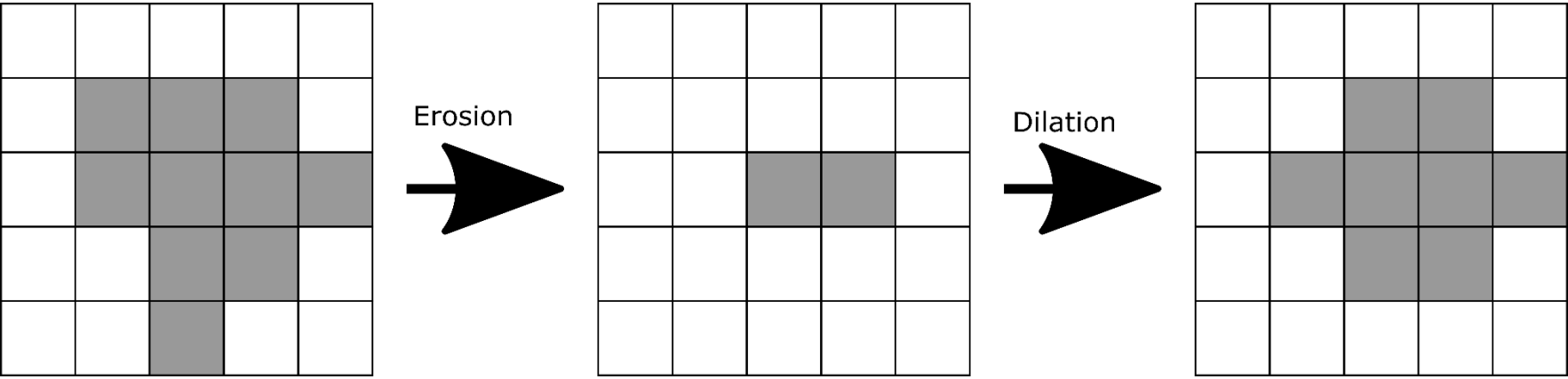
Die **Schliessung** einer Menge X mit einem Strukturelement B ist definiert als

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

Morphologische Operationen

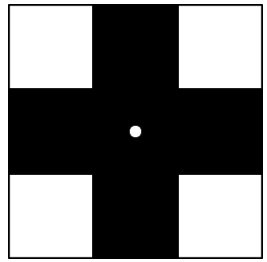
Öffnung / opening

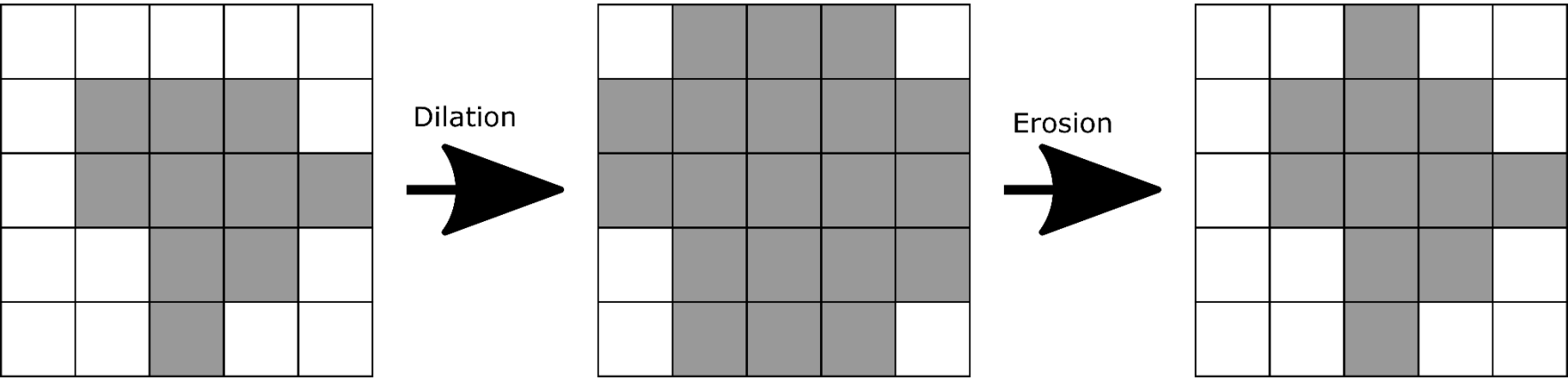




Morphologische Operationen

Schließung / Abschluss / closure





Morphologische Operationen

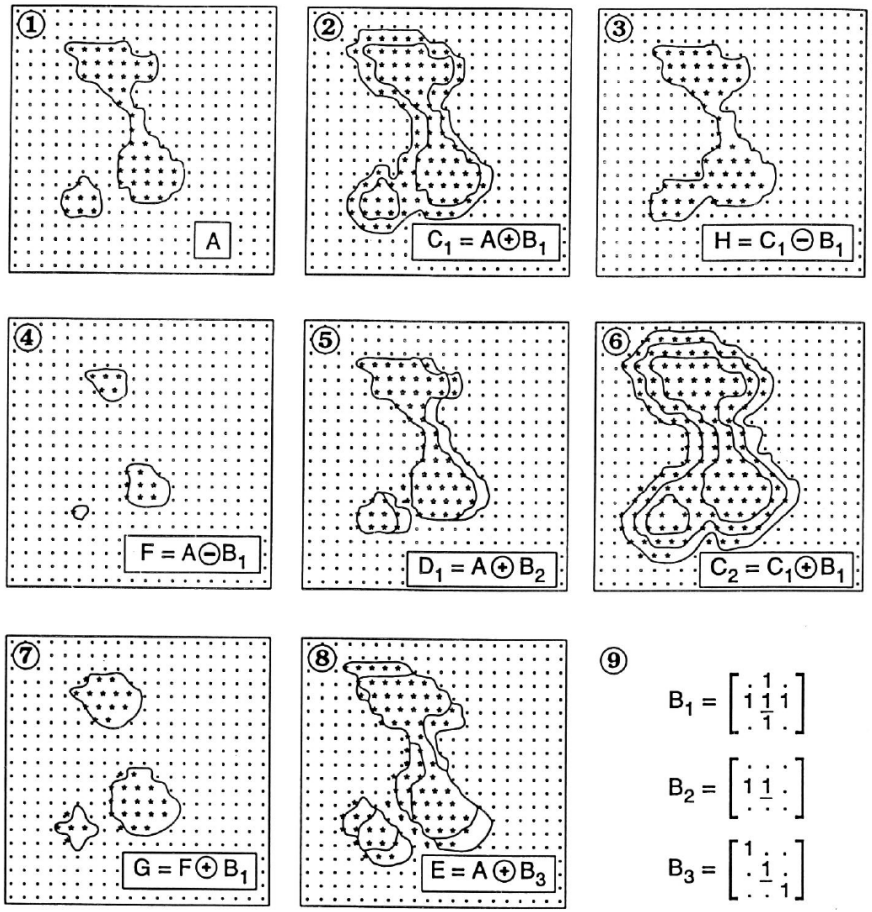


FIG. 6-9. Mathematical morphology operations on a small raster image. **(1).** Original image **A**. **(2).** Dilation of **A** using structuring element B_1. **(3).** Closing of image **A** by B_1. **(4).** Erosion of **A** by B_1. **(5).** Dilation of **A** by structuring element B_2. **(6).** Two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

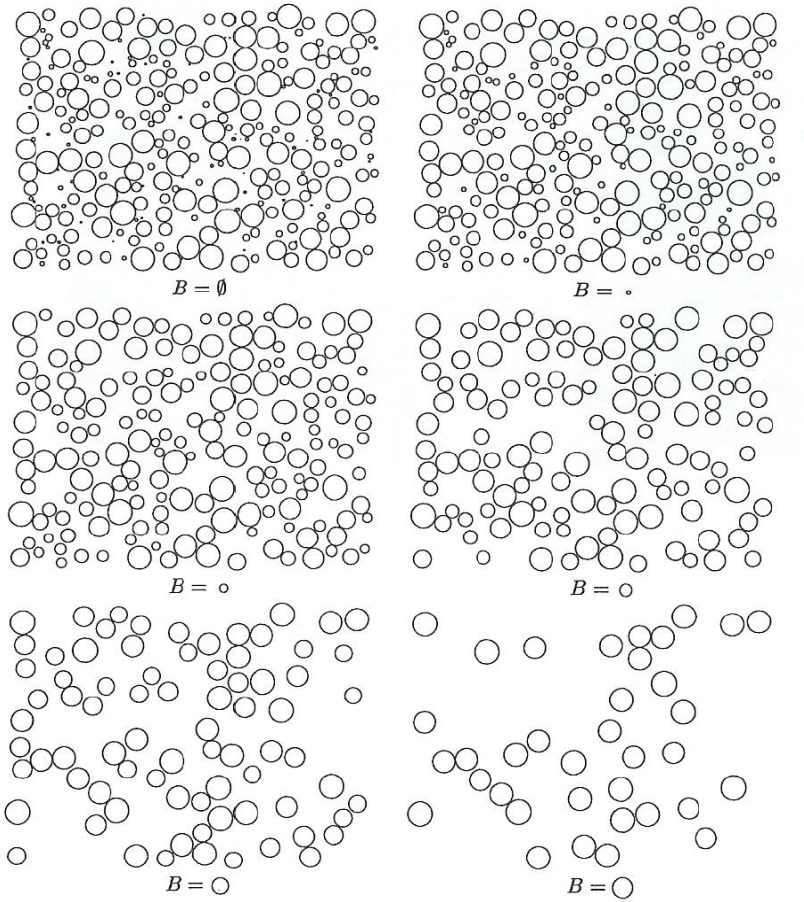


Figure 2.3 'Sieving' of disks of uniformly distributed diameter. The structuring element B is a disk of increasing diameter. If the diameter of the disk is smaller than the diameter of the current structuring element, this disk is eliminated from the image. For $B = \emptyset$ we have the original image.

Morphologische Operationen

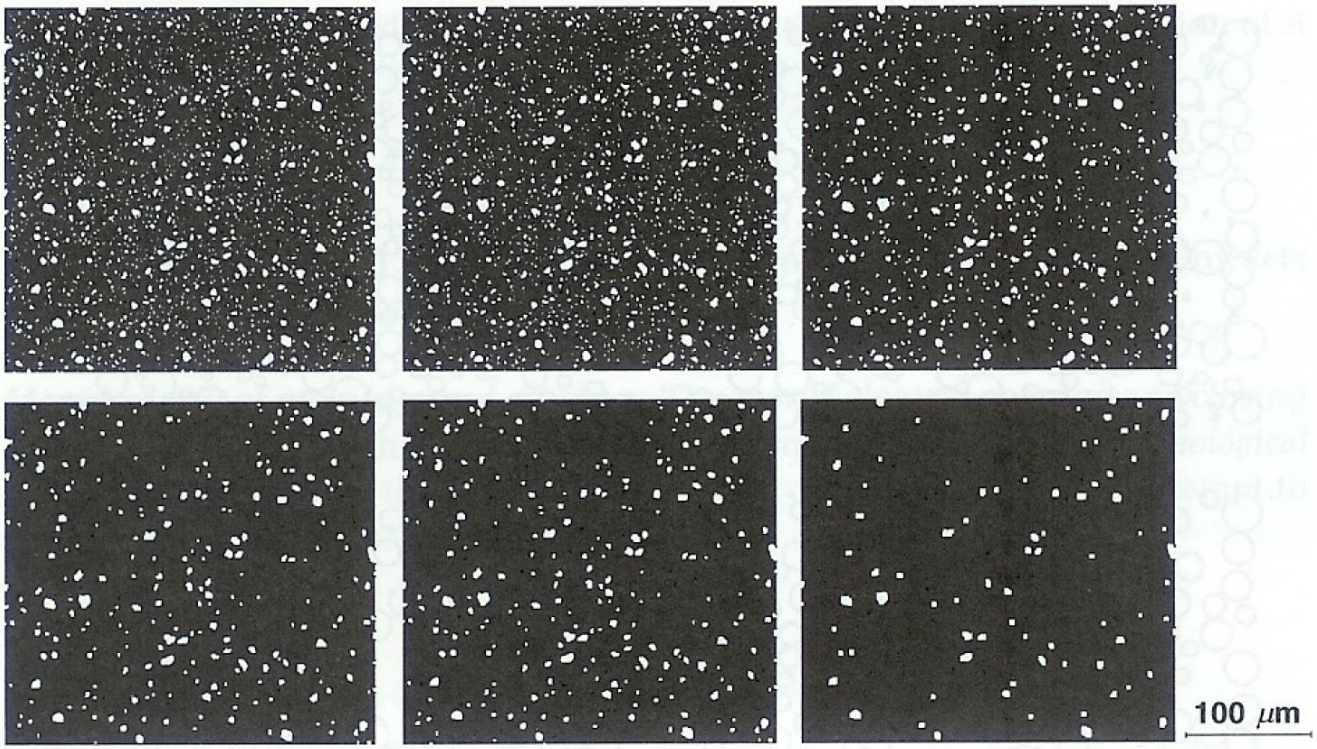
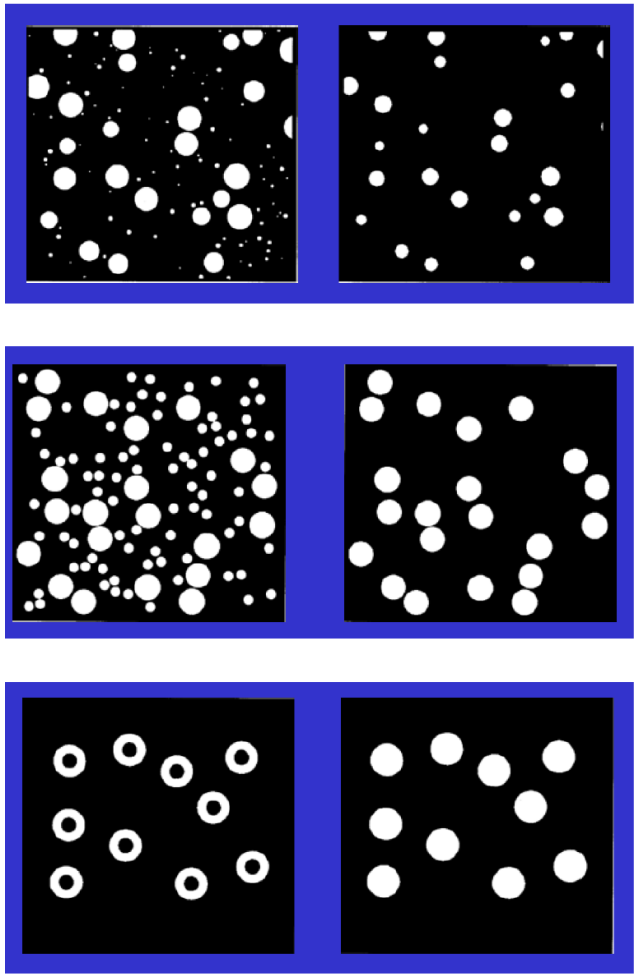


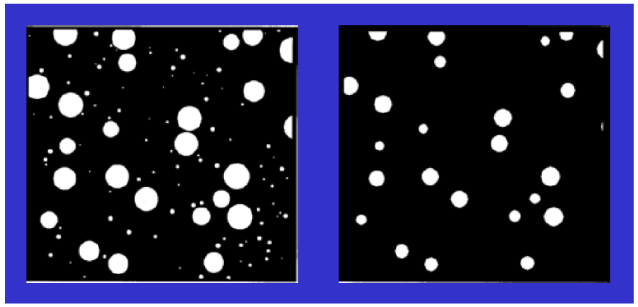
Figure 2.4 The use of morphological opening for the classification of the particle sizes of a spheroidized pearlite containing small granular carbide particles. The structuring element is a square of increasing edge length. This example shows that the shape of the objects is changed during the opening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

Morphologische Operationen

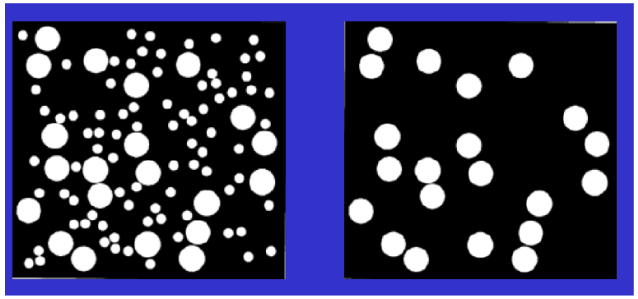


Welche Operationen wurden durchgeführt?

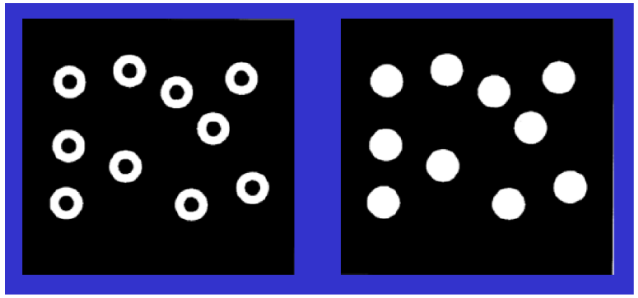
Morphologische Operationen



Erosion

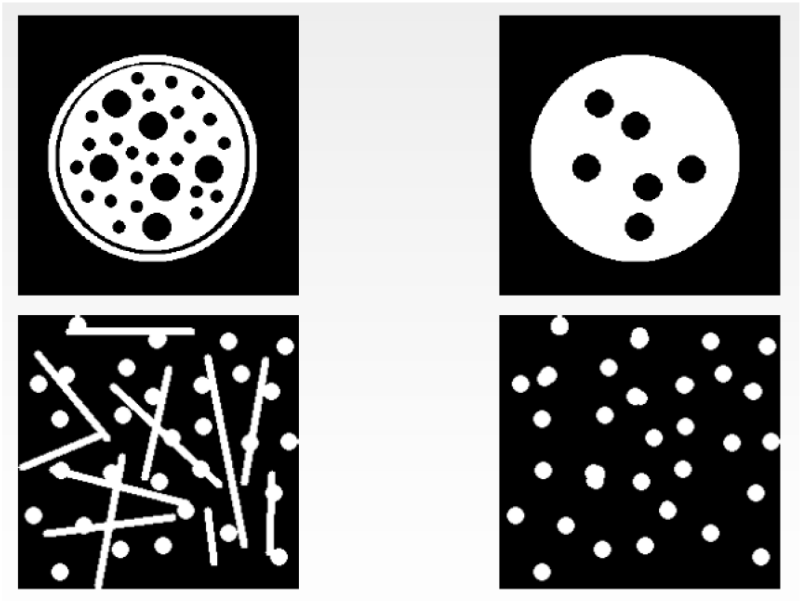


Öffnung



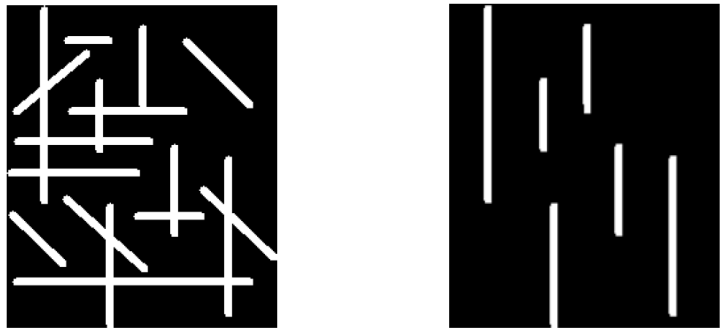
Schliessung

Morphologische Operationen

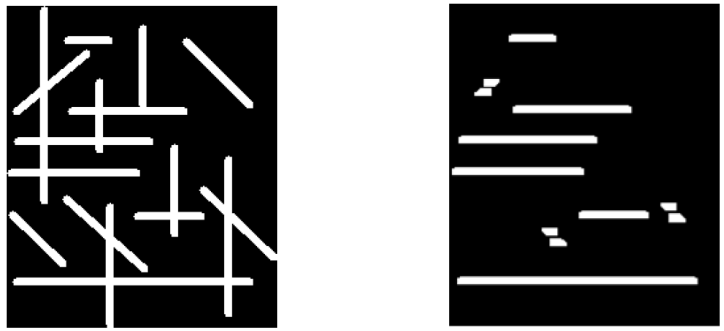


top: Closing with a disc shaped structuring element
bottom: Opening with a disk shaped structuring element

Morphologische Operationen



Opening with a 9×3 vertically oriented structuring element.



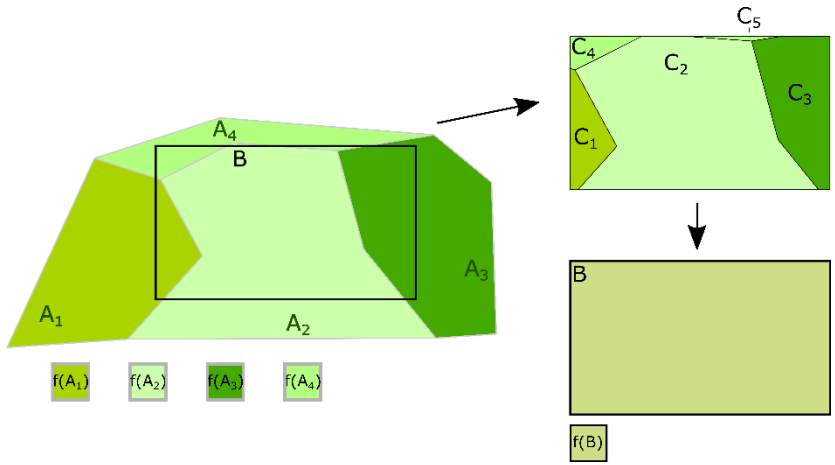
Opening with a 3×9 horizontally oriented structuring element.

Sampling Transformationen

Probleme

- Gegeben 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*)



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

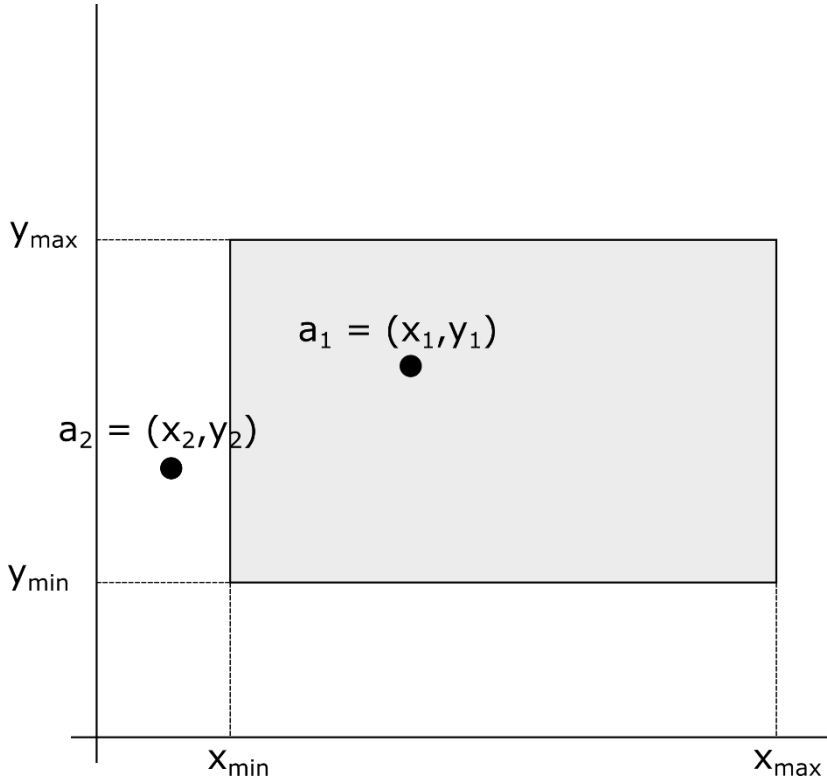
- Gegeben:
  Rechteck $R = (x_{min}, x_{max}, y_{min}, y_{max})$
  Abfragepunkt (x, y)

- 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

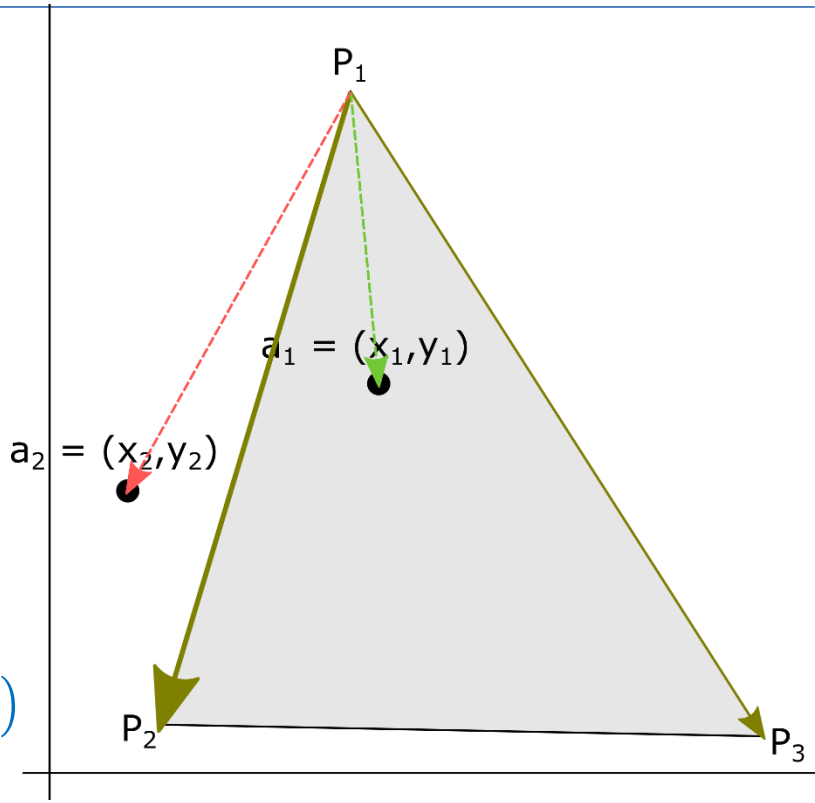
- Gegeben:
  Dreieck $\Delta(\vec{\boldsymbol{p}}_i, \vec{\boldsymbol{p}}_j, \vec{\boldsymbol{p}}_k)$
  Abfragepunkt $\vec{\boldsymbol{x}} = (x, y)$

- 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

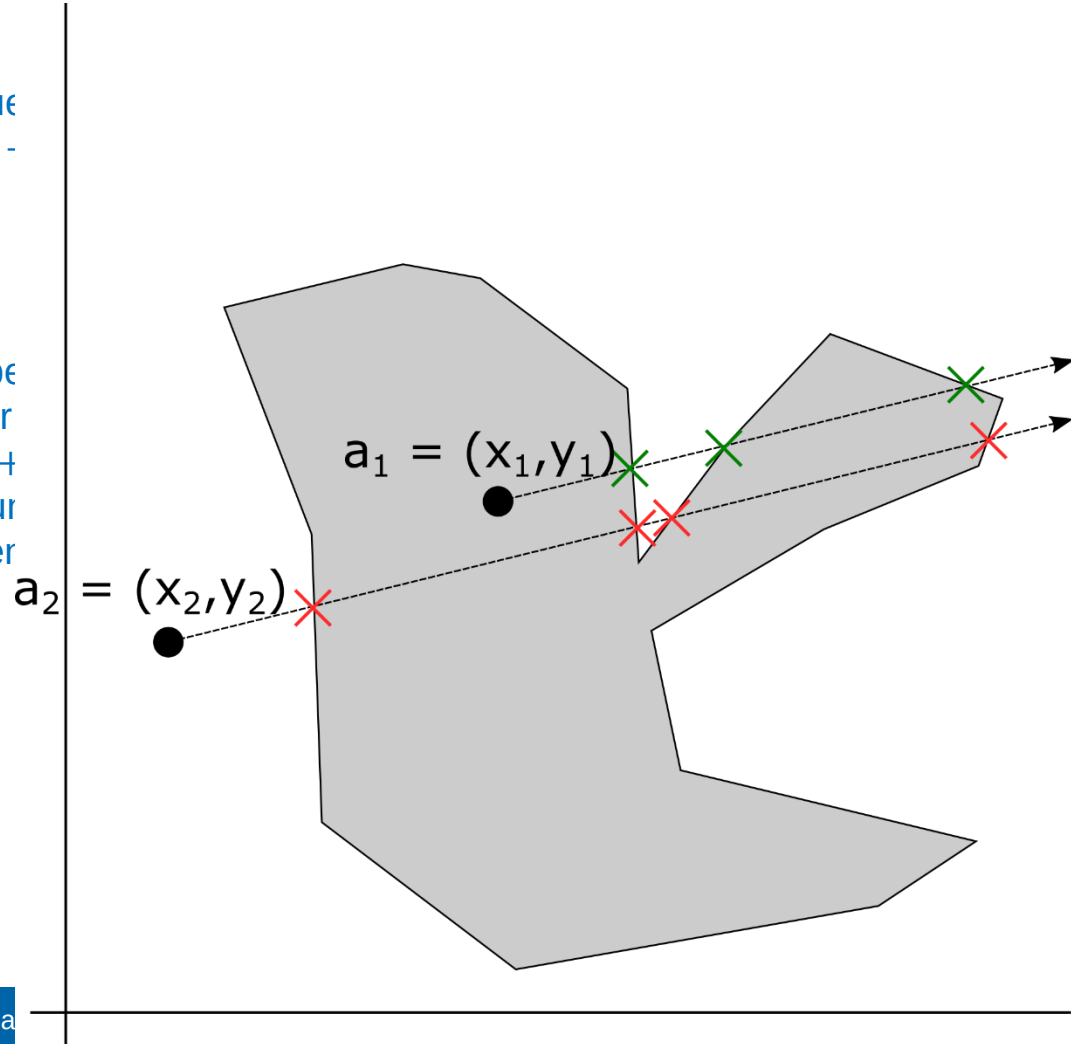
- Gegeben:
  Polygon $\boldsymbol{P}(\vec{\boldsymbol{p}}_1, \vec{\boldsymbol{p}}_2, \ldots, \vec{\boldsymbol{p}}_n)$ mit Seque
  $\Rightarrow$ Kante $k_i = \vec{\boldsymbol{p}}_{i+1} - \vec{\boldsymbol{p}}_i, 1 \leq i \leq n -$

  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



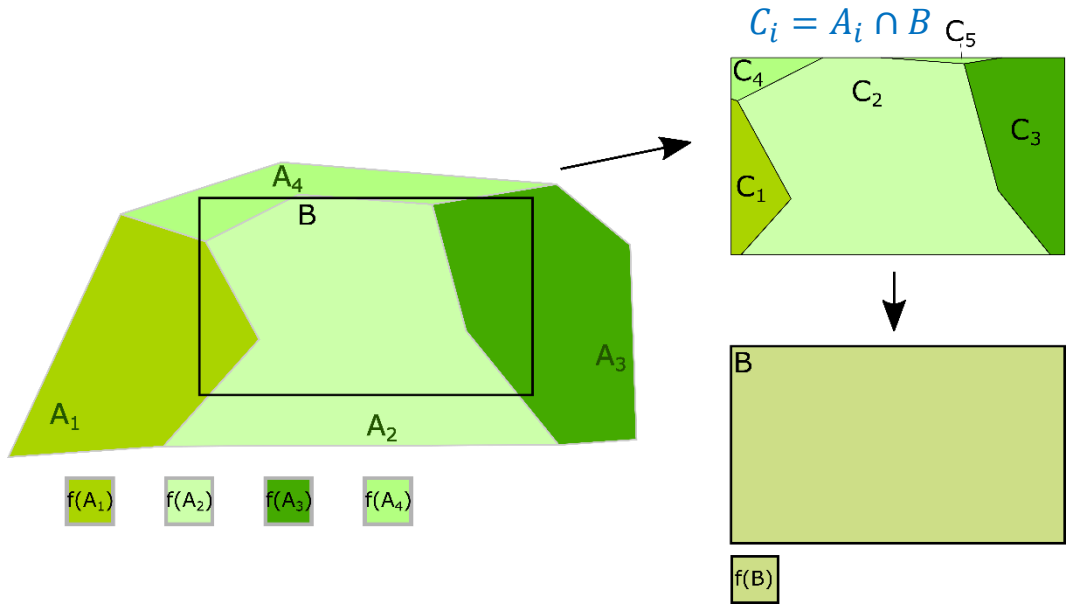
Fläche-zu-Fläche Transformation

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

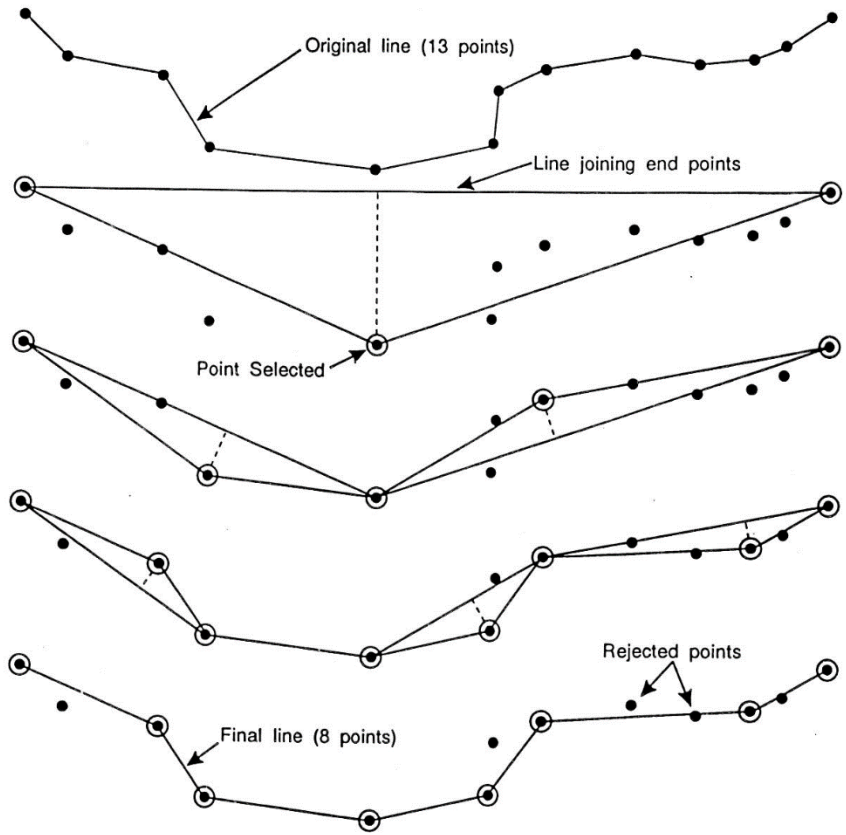


FIG. 6-10. Douglas-Peucker algorithm for line generalization. The original line contains 13 points, and the final line contains 8 points. The rejected points are said to be "weeded". A number of alternative algorithms for line weeding have been proposed.