

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)

Räumliche 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 werden (Höhen, Stoffkonzentrationen, ...)
=> vergl. Skalen-Begriff aus "Datenanalyse/Statistik"

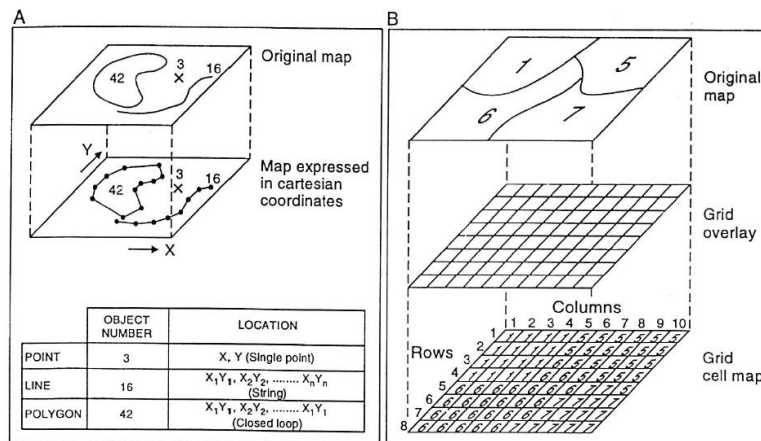
Objekttypen

- **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 Stoffkonzentrationen 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
- grundlegene Einheiten geben Auflösung vor



Raster- vs. Vektormodell

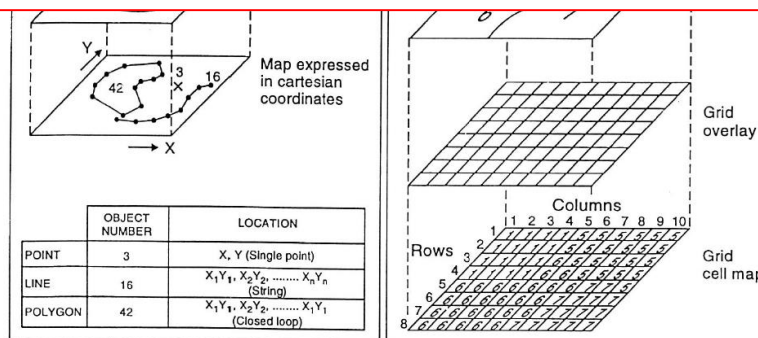
- **Rastermodell:**

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

- **Vektormodell:**

- Grundlegende 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

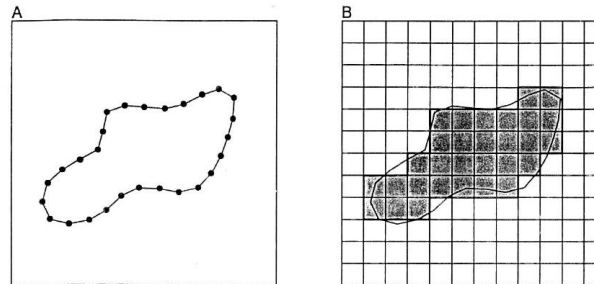
Sagen Sie Bitte niemals, einfach nur: „die Grundelemente eines Vektormodells sind Vektoren“!



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



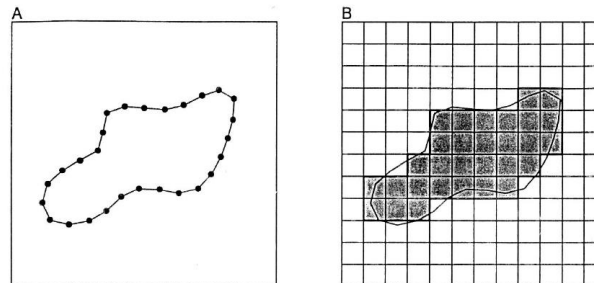
Raster- vs. Vektormodell

- **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

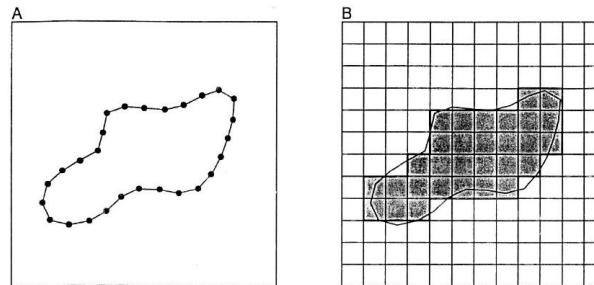
- **Rastermodell:**

- 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

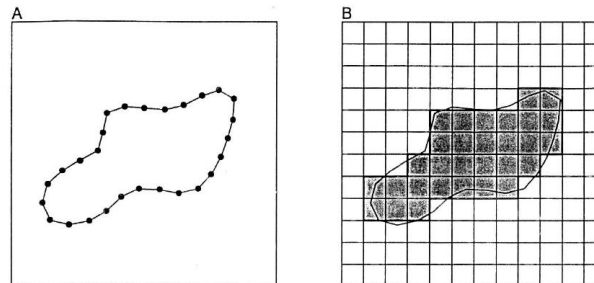
Bestimmung von Umfang und Fläche eines Objektes

- **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



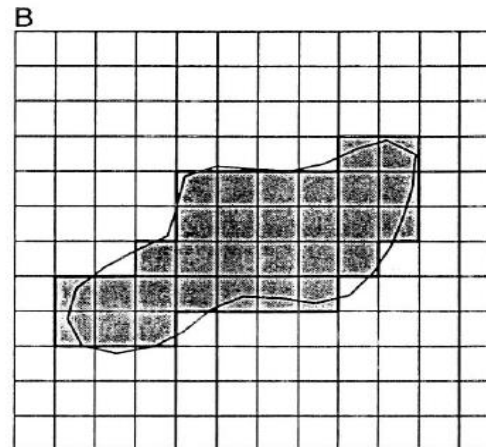
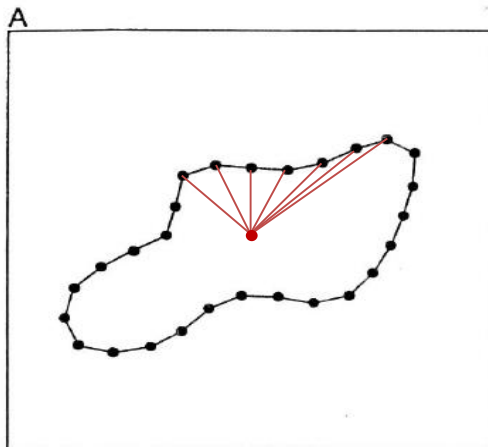
Bestimmung von Umfang und Fläche eines Objektes

- **Rastermodell:**

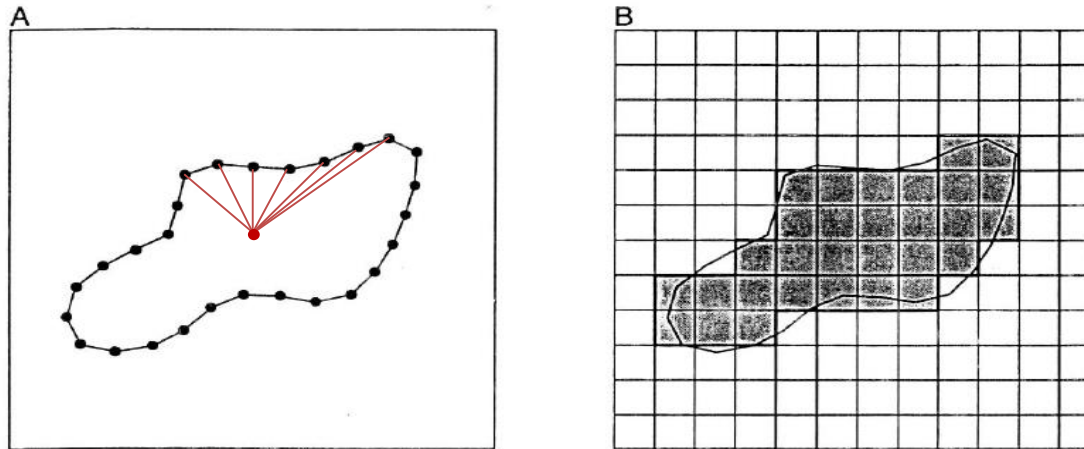
- Fläche = Anzahl Pixel x Fläche pro Pixel
- Umfang = Anzahl der Randpixel x Länge einer Pixelkante

- **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

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

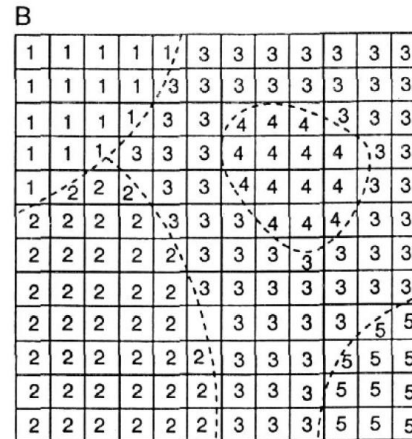
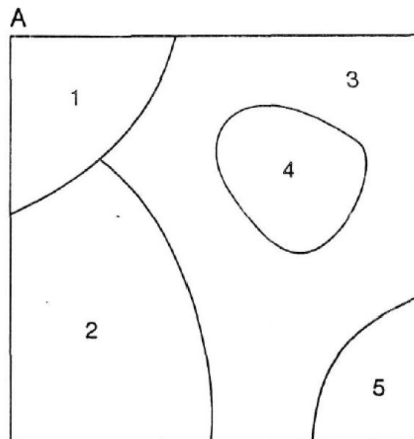
Beachte, dass für letzteres Aufhebungseffekte bei der Summierung über alle Dreiecke eine Rolle spielen, so dass der Referenzpunkt (x, y) herausfällt.

Raster- vs. Vektormodell

Geoobjekte können über verschiedene Attribute beschrieben werden:

- räumliche Attribute
- zeitliche Attribute
- thematische Attribute

Die Attribute können
Möglichkeit der



hinterlegt werden.

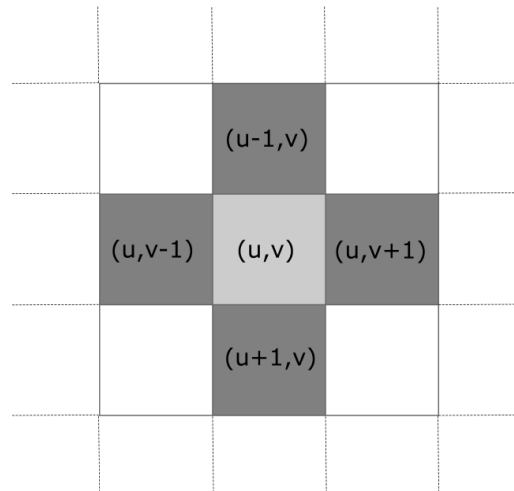
C

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

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 werden.

Nachbarschaftsbeziehungen sind einfach über die benachbarten Pixel zu definieren.

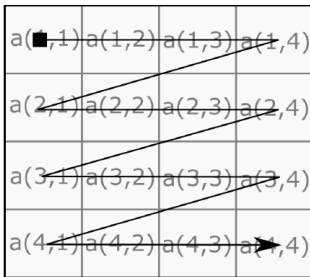
Wie werden die einzelnen Pixel (und damit auch Nachbarschaftsbeziehungen) angesteuert? Eine entsprechende **Datenstruktur** ist notwendig.

Kann einfach zwischen verschiedenen Layern/Bändern gewechselt werden?

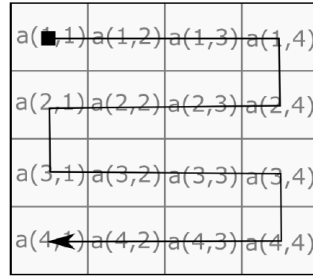
		Spalte (<i>column</i>)							
		1	2	3	4				
Zeile (<i>row</i>)	1	a(1,1)	a(1,2)	a(1,3)	a(1,4)	b(1,1)	b(1,2)	b(1,3)	b(1,4)
	2	a(2,1)	a(2,2)	a(2,3)	a(2,4)	b(2,1)	b(2,2)	b(2,3)	b(2,4)
	3	a(3,1)	a(3,2)	a(3,3)	a(3,4)	b(3,1)	b(3,2)	b(3,3)	b(3,4)
	4	a(4,1)	a(4,2)	a(4,3)	a(4,4)	b(4,1)	b(4,2)	b(4,3)	b(4,4)
		Band A				Band B			

Rastermodell

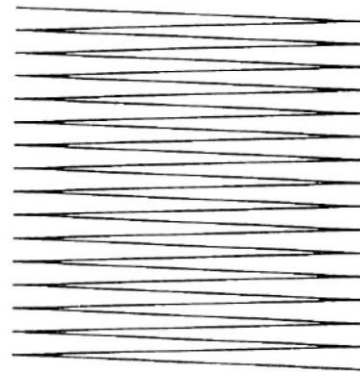
Oft ist es effizient und einfach Matrixeinträge sequentiell in einer Zeile abzulegen. Dann muss jedoch die Durchlaufvorschrift der Matrix bekannt sein!



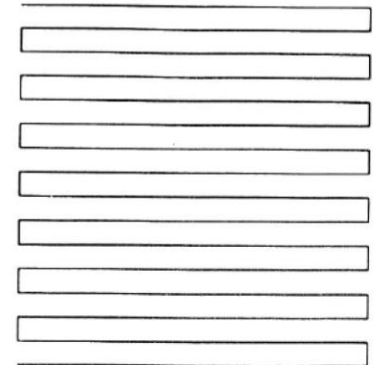
Band A



Band A



(a) Row Order



(b) Row-prime Order



(c) Morton Order



(d) Pi-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