TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG

Die Ressourcenuniversität. Seit 1765.

Grundlagen GIS

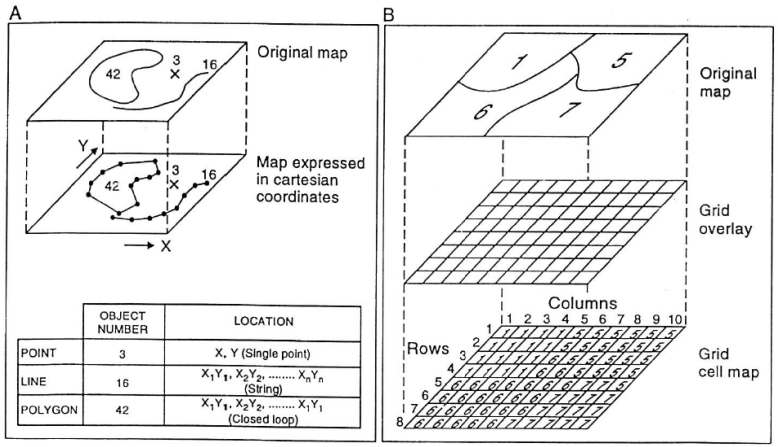
Sommersemester 2024

Prof. Christian Gerhards
Arbeitsgruppe Geomathematik und Geoinformatik

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

Raster- vs. Vektormodell

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



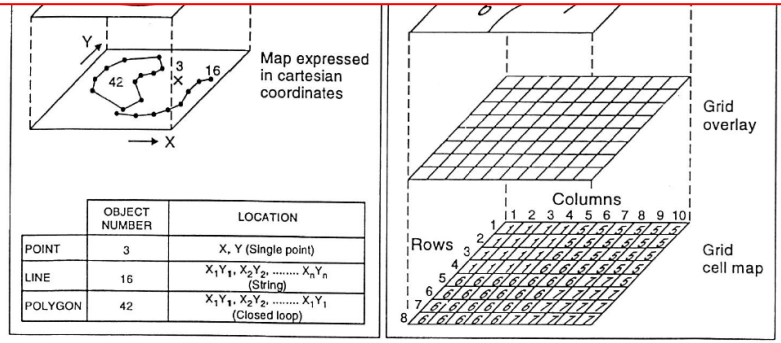
	OBJECT NUMBER	LOCATION
POINT	3	X, Y (Single point)
LINE	16	$X_1Y_1, X_2Y_2, \ldots\ldots X_nY_n$ (String)
POLYGON	42	$X_1Y_1, X_2Y_2, \ldots\ldots X_1Y_1$ (Closed loop)

Raster- vs. Vektormodell

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

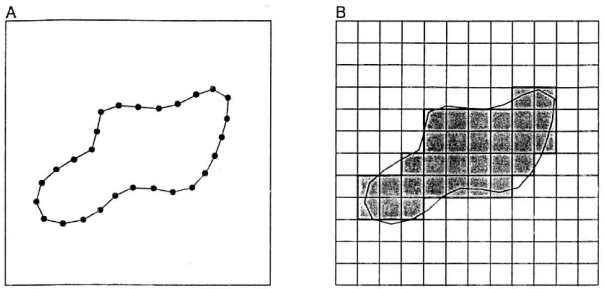
- **Vektormodell**:
  - Grundelegende Einheiten sind Vertexe (0-D Punktobjekte); und darauf aufbauend Linien/Polygonzüge, Polygone, …
  - Exakte Koordinaten für Vertexe, kein Auflösungsverlust
  - jedem Element kann ein oder mehrere Attributwerte zugeordnet werden

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



Raster- vs. Vektormodell

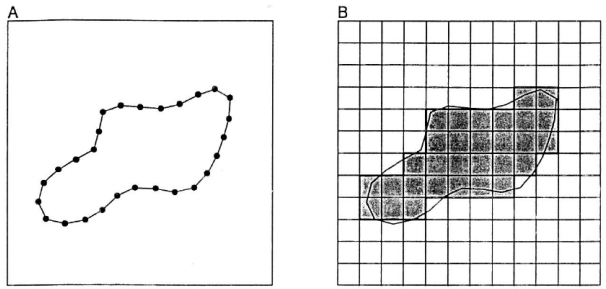
- **Rastermodell**:
  - ausschliesslich flächenhafte Betrachtung von Objekten: jedes Pixel beschreibt eine (möglicherweise sehr kleine) Fläche
  - größere Einzelobjekte lassen sich nur über Attributwerte voneinander abgrenzen
  - Datenerfassung einfach und effizient
  - Position und Nachbarschaftsbeziehungen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



Raster- vs. Vektormodell

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

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



Raster- vs. Vektormodell

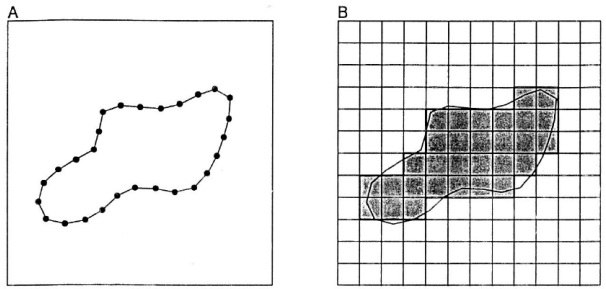
Bestimmung von Umfang und Fläche eines Objektes

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

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

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

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

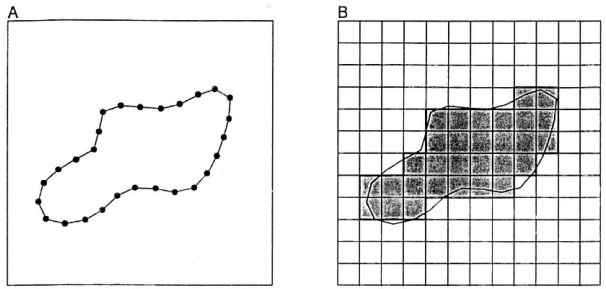


Raster- vs. Vektormodell

Bestimmung von Umfang und Fläche eines Objektes

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

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

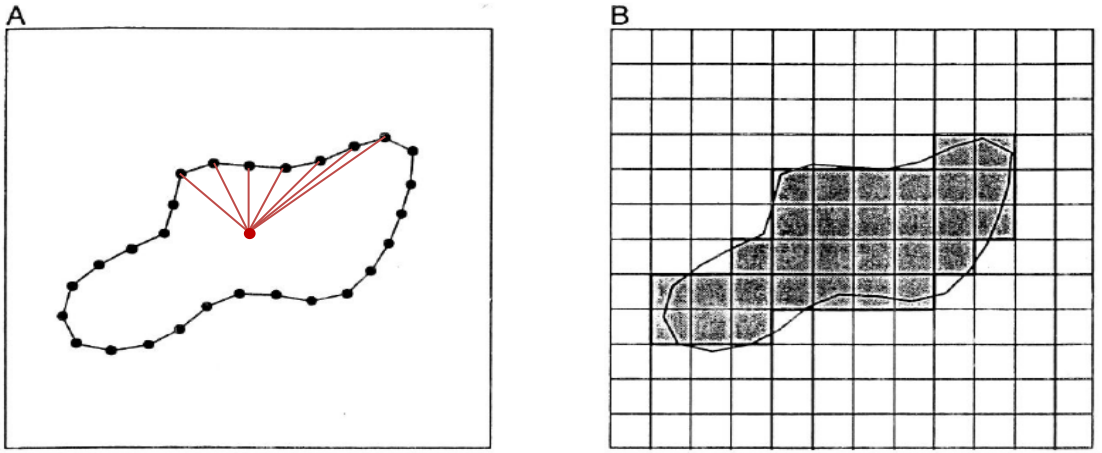


Raster- vs. Vektormodell

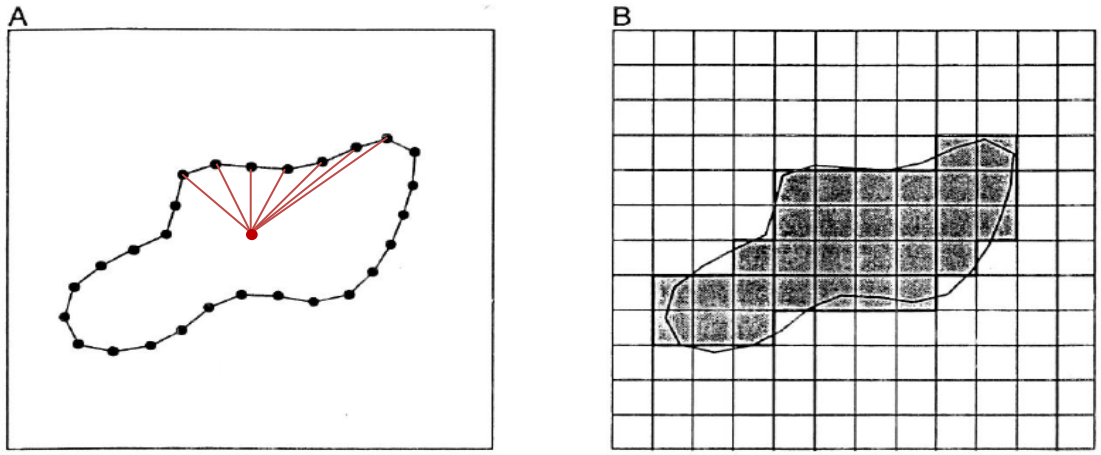
Bestimmung von Umfang und Fläche eines Objektes

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

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



Raster- vs. Vektormodell



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

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

Für das gesamte Polygon im Vektormodell ergibt sich als Fläche

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

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

Raster- vs. Vektormodell

Geoobjekte können über verschiedene Attribute beschrieben werden:

- räumliche Attribute
- zeitliche Attribute
- thematische Attribute

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

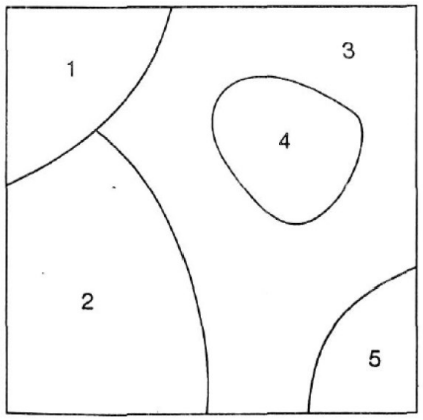
Raster- vs. Vektormodell

Geoobjekte können über verschiedene Attribute beschrieben werden:

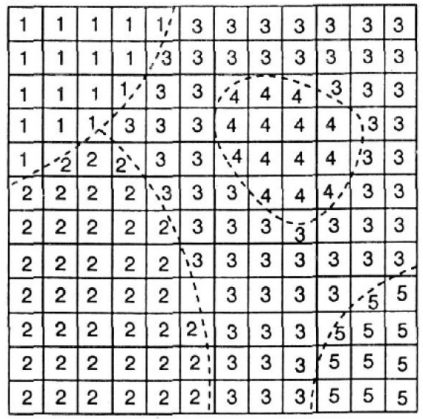
- räumliche Attribute
- zeitliche Attribute
- thematische A

Die Attribute k hinterlegt werden. Möglichkeit der \

A



B



C

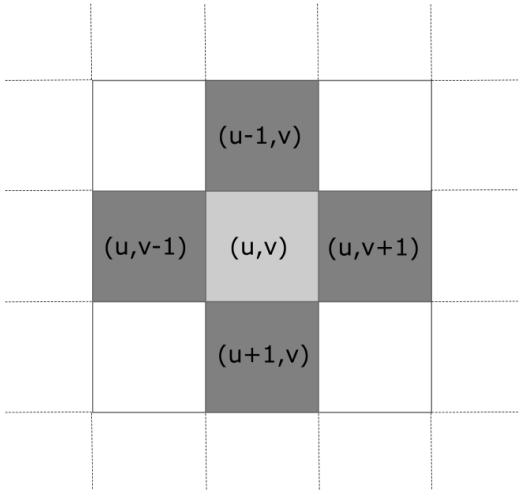
Polygon	Class	Rock type	Age	Name
1	15	sandstone	Late Pennsylvanian	Andrews Formation
2	6	limestone	Early Silurian	Barry Formation
3	3	shale	Middle Silurian	Clinton shale
4	14	granite	Devonian	Delta granite
5	14	granite	Devonian	Delta granite

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

Rastermodell

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

Nachbarschaftsbeziehungen sind einfach über die benachbarten Pixel zu definieren.



Rastermodell

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

Nachbarschaftsbeziehungen sind einfach über die benachbarten Pixel zu definieren.

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

Kann einfach zwischen verschiedenen Layern/Bändern gewechselt werden?

Spalte (*column*)

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

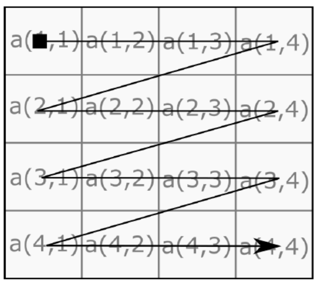
Band A

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

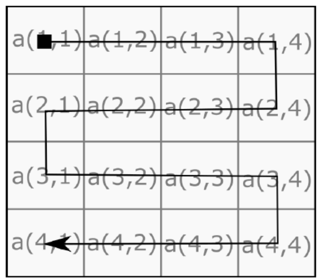
Band B

Rastermodell

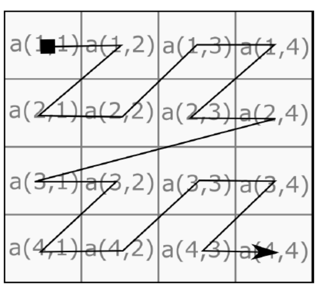
Oft ist es effizient und einfach, Matrixeinträge sequentiell in einer Zeile abzulegen. Dafür muss jedoch die Durchlaufvorschrift der Matrix bekannt sein!



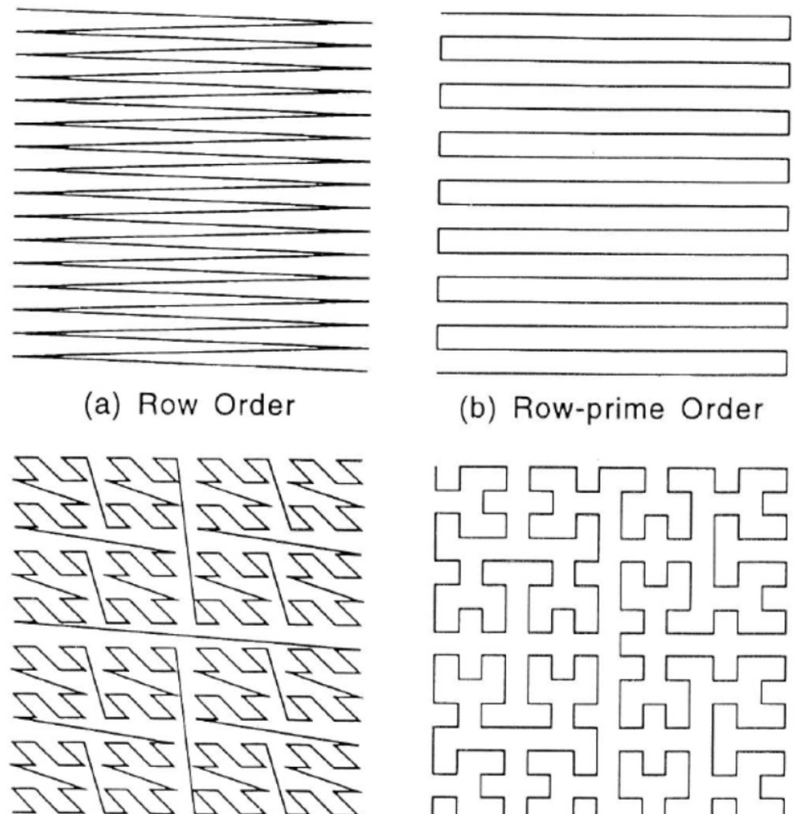
Band A



Band A

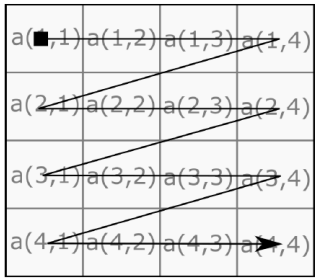


Band A

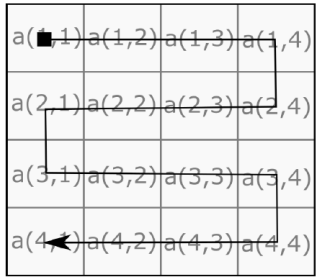

(a) Row Order
(b) Row-prime Order
(c) Morton Order
(d) Pi-Order

Rastermodell

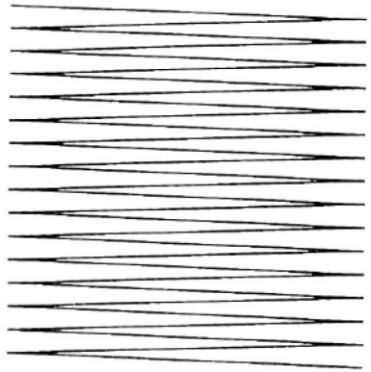
Oft ist es effizient und einfach, Matrixeinträge sequentiell in einer Zeile abzulegen. Dafür muss jedoch die Durchlaufvorschrift der Matrix bekannt sein!

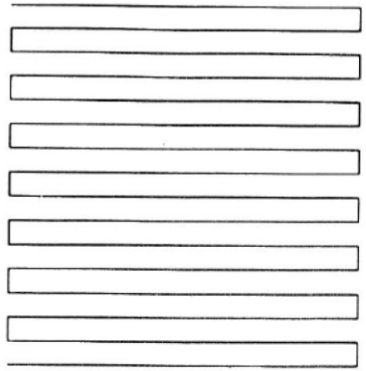


Band A



Band A

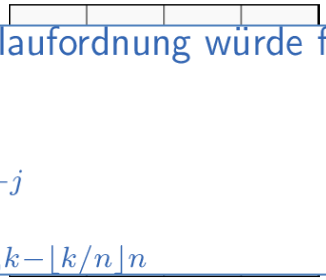

(a) Row Order

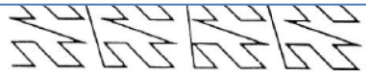

(b) Row-prime Order

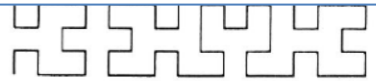
Welcher Durchlaufordnung würde folgende Zuweisung von Zeileneinträgen s_k zu Matrixeinträgen $a_{i,j}$ entsprechen?

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

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


Band A


(c) Morton Order

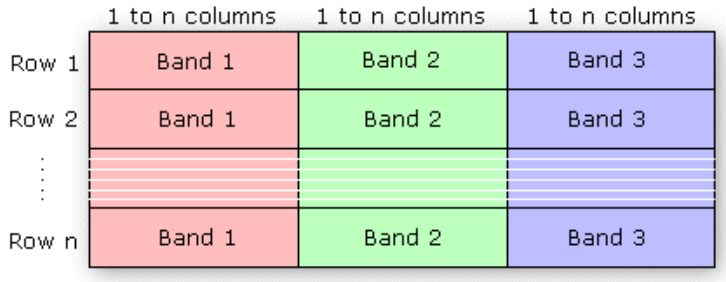

(d) Pi-Order

Rastermodell

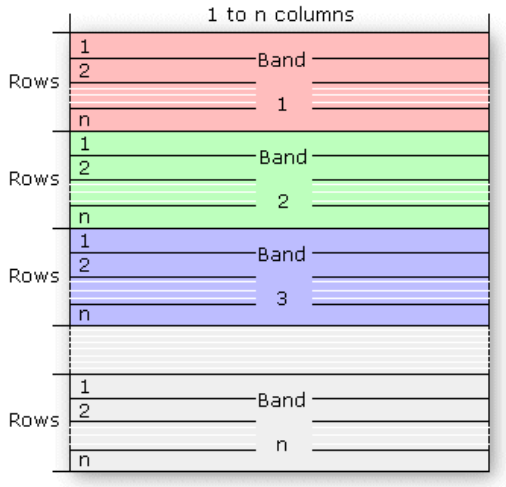
Angenommen wir wollen mehrere Bänder/Layer durchlaufen (z.B. bei hyperspektralen Daten). Welche Durchlaufvorschriften gibt es dann?

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

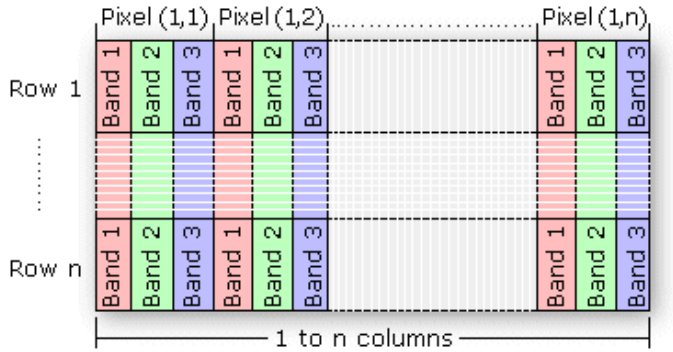
Band Interleaved by Line (BIL)



Band sequential (BSP)



Band Interleaved by Pixel (BIP)



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

Rastermodell

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

Hier wollen wir zwei einfache Methoden besprechen:

- **Lauflängenkodierung**

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

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

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

Rastermodell

- **Blockkodierung**

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

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

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

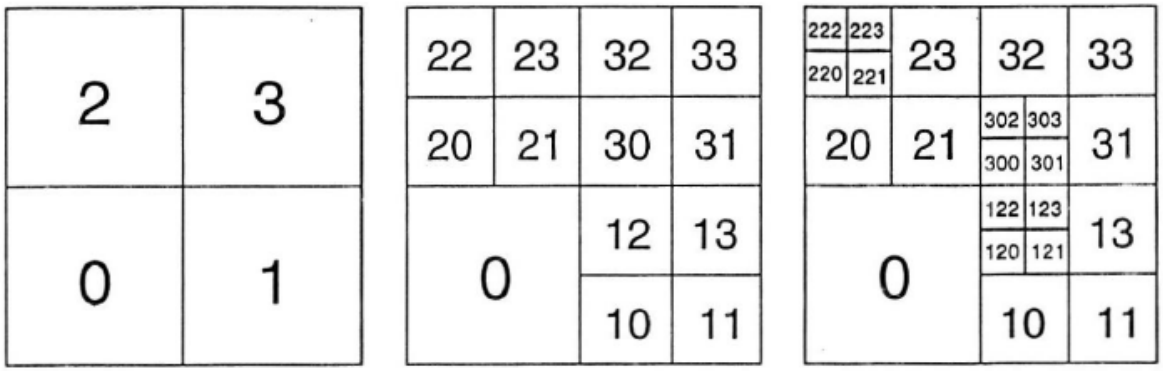
Die ersten zwei Einträge beschreiben die Zeile und Spalte der linken oberen Ecke eines Quadrates, der dritte Eintrag die Seitenlänge des Quadrates, der vierte Eintrag das Attribut.

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

Rastermodell

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

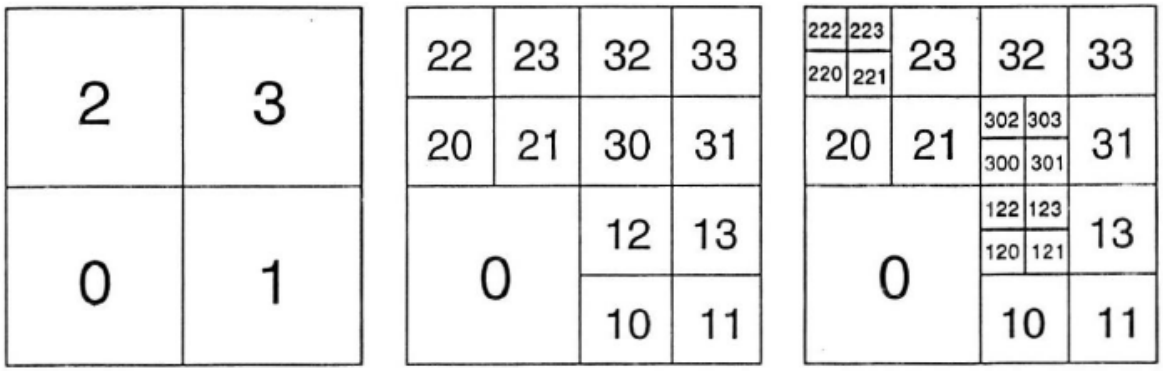
Ein weiteres auf flächige homogene Objekte abzielendes System. Sukzessive, hierarchische Unterteilung des Rasters in kleiner werdende Quadranten.



Rastermodell

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

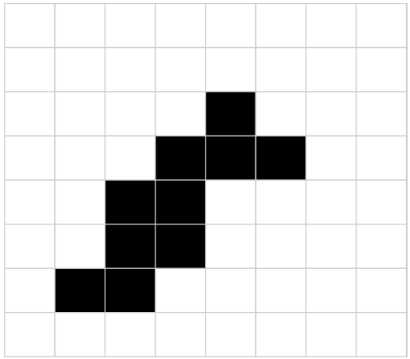
Ein weiteres auf flächige homogene Objekte abzielendes System. Sukzessive, hierarchische Unterteilung des Rasters in kleiner werdende Quadranten.

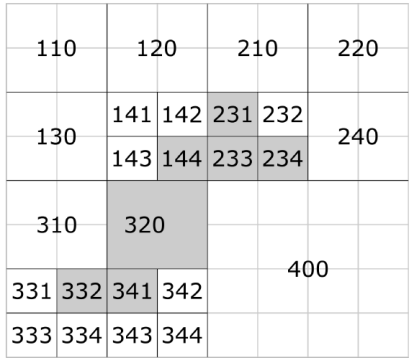


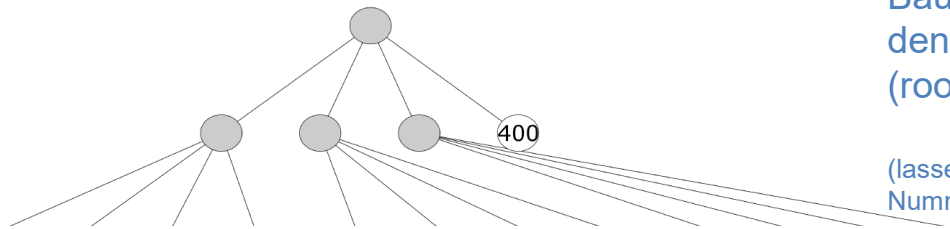
Es bietet sich an die Quadranten mit einem Zahlsystem zur Basis vier zu versehen, für eine intuitive Nummerierung der Quadranten.

Rastermodell

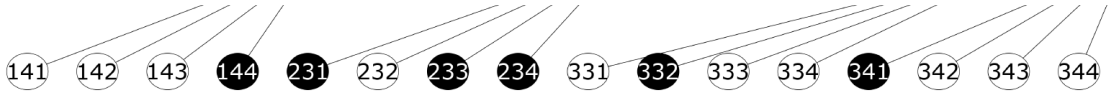
- **Quadtrees (Octrees in 3-D)**







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



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

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

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

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

Wie verhält sich Speicheraufwand von Quadtrees zur vorher beschriebenen Blockkodierung?