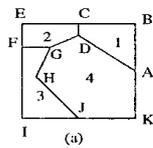


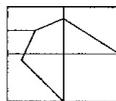
(Raster-)Datenhaltung

Rasterdaten

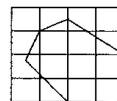


(a)

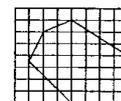
Regelmäßiges Raster:



(b)

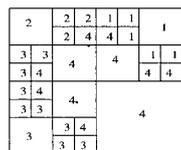


(c)



(d)

Optimiertes Raster: **Quadtree**

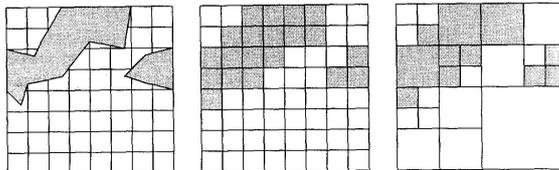


(e)

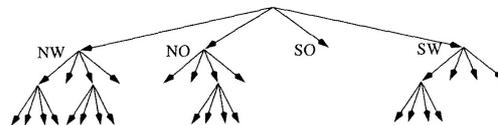
(Adam und Gangopadhyay 1997)

Quadtree

Rekursive Aufteilung in vier Quadranten:



Resultierende Baumstruktur:



(Bartelme 2003)

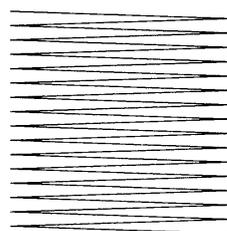
Codierungsformate

- **JPEG** = *Joint Photographic Experts Group* (urspr. *JFIF = JPEG File Interchange Format*)
- **GIF** = *Graphic Interchange Fomat* (CompuServe)
- **PNG** = *Portable Network Graphics* (Public Domain)
- **TIFF**: *Tag Image File Format*: definiert *Tags* (Etiketten) für versch. Arten der Kodierung
- **GeoTIFF**: Erweiterung des TIFF mit Metadaten über verwendete Bezugssysteme => automatische Georeferenzierung

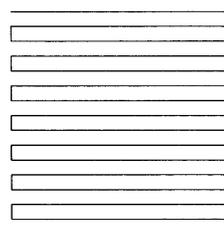
Speichern von Rasterdaten

- Speicherung von Attributwerten, z.B. Grauwerten
- z.B. **Quadtree**-Unterteilung von Rastern
- Alternativ: **Laufängencodierung**: Nachbarzellen weisen sehr oft den gleichen Wert auf => geringer Speicherbedarf, wenn bei Zeilen- (Spalten-)weisem Durchlauf nur die Änderung des Wertes gespeichert wird
- **Differentielle Laufängencodierung**: Speicherung der Grauwertdifferenz
- Statt Zeilen- oder Spalten-weisem Vorgehen: raumfüllende Kurven oder Indizierung

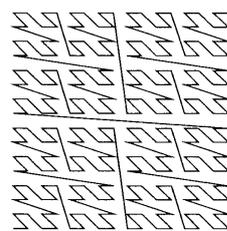
Raumaufteilung (Linien)



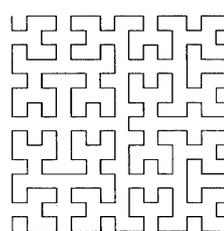
(a) Row Order



(b) Row-prime Order



(c) Morton Order



(d) Pi-Order

(Bonham-Carter 1994)

Raumaufteilung (Linien)

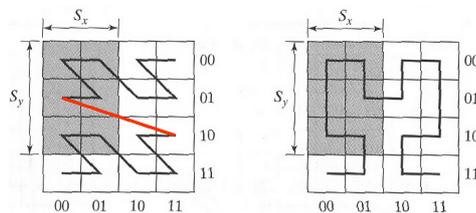
Problem: 2D-(Raster-)Daten sollen in einem 1D-Speichermedium so angeordnet werden, dass Nachbarschaftsbeziehungen weitgehendst erhalten bleiben

=> Anordnung der Rasterpunkte in einer **raumfüllenden Kurve** (vergl. Fraktale):

- **Z-Kurve = Peano-Kurve**
- **Hilbert-Kurve**

Z-Kurve vs. Hilbert-Kurve

- Hilbert-Algorithmus ist aufwendiger
- i.d.R. Zugriffszeiten der Hilbert-Kurve etwas geringer, da im Gegensatz zur Z-Kurve die Hilbert-Kurve generell nur Punkte verbindet, die auch im 2-D-Raum benachbart sind:

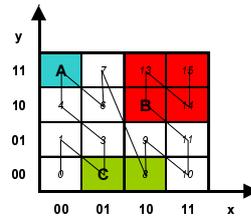


→ allgemeines Maß für die Güte der Zerlegung:

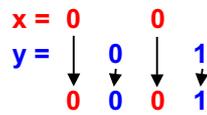
Bestimmung der Aufteilung einzelner Objekte in unterschiedliche Cluster

Algorithmen: Z-Kurve (Peano-Kurve)

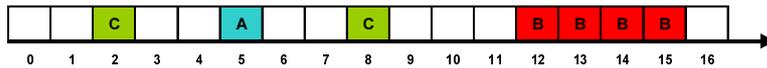
1. Einlesen der x- und y-Koordinaten (Binärwerte)



2. Vermengen der Koordinaten

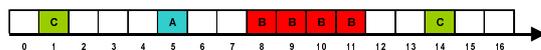
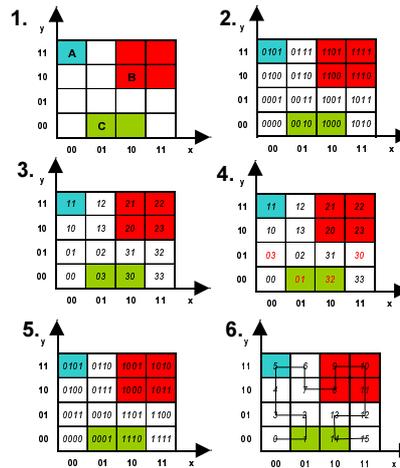


3. Berechnung des Z-Wertes (Dezimalwert)



Algorithmen: Hilbert-Kurve

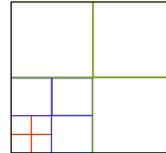
- Einlesen der x- und y-Koordinaten (Binärwerte)
- Vermengen der Koordinaten
- Aufteilung des Strings von links nach rechts in 2-bit-Strings; ersetze „00“ durch „0“, „01“ durch „1“, „10“ durch „3“, „11“ durch „2“
- Ersetze jeweils „01“ durch „03“, „03“ durch „01“, „30“ durch „32“ und „32“ durch „30“
- Umrechnung der 2-bit-Strings in Binärwerte
- Zusammensetzung der Binärwerte, und Berechnung der Z-Werte (Dezimalwert)



Z-Kurve: Räumliche Bereiche

Indizierung: Objekte werden unter den z-Werten ihrer Punkte/Rasterzellen abgelegt

Räumliche Bereiche lassen sich durch ≥ 1 quadratische Blöcke darstellen, die sich als durch **Quadtree**-Unterteilungen der Fläche entstanden auffassen lassen:

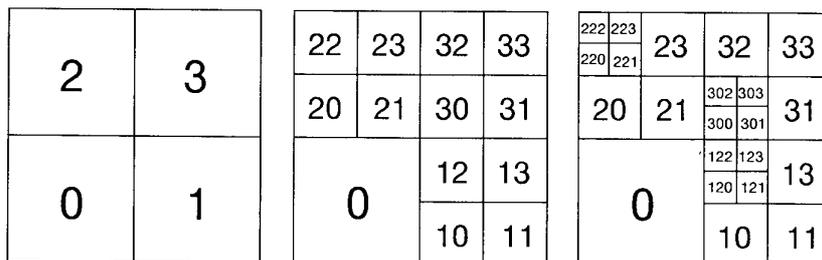


=> Jedes Objekt kann eindeutig durch die z-Werte seiner Blöcke (bzw. min. und max. z-Werte) dargestellt werden

Die z-Werte der Rasterzellen der Blöcke haben jeweils das gleiche **Präfix** (in Binärwerten), z.B. **11**** (**1100**, **1101**, **1110**, **1111**)

=> Identifizierung der Blöcke, und damit auch der Objekte

Z-Kurve: Räumliche Bereiche



(Bonham-Carter 1994)

Die z-Werte der Rasterzellen der Blöcke haben jeweils das gleiche **Präfix** (in Binärwerten), z.B. **11**** (**1100**, **1101**, **1110**, **1111**)

=> Identifizierung der Blöcke, und damit auch der Objekte

Z-Kurve: Abfragen

Suche nach:

Punkten: suchen in dem nach z-Werten sortiertem File, oder B-Baum-Suche

Bereichen: für 2 Blöcke mit den Präfixen z_1 und z_2 , $z_1 < z_2$, gilt: wenn z_1 das Präfix von z_2 ist, enthält der Bereich r_1 den Bereich r_2 ; ansonsten sind r_1 und r_2 räumlich disjunkt

dem nächstem Nachbarn zum Punkt P_i ; berechne den z-Wert des Punkts P_i , suche den Punkt P_j mit dem nächsten z-Wert, und führe eine räumliche Abfrage mit dem Radius $r = \text{Abstand der beiden Punkte}$ durch

Räumliche Verschneidungen: Sortiere jeweils die z-Werte der Gruppen der zu verschneidenden Objekte, verschneide die beiden Listen