

Datastrukturer och modeller

Thomas Gumbricht
thomas@karttur.com
www.karttur.com

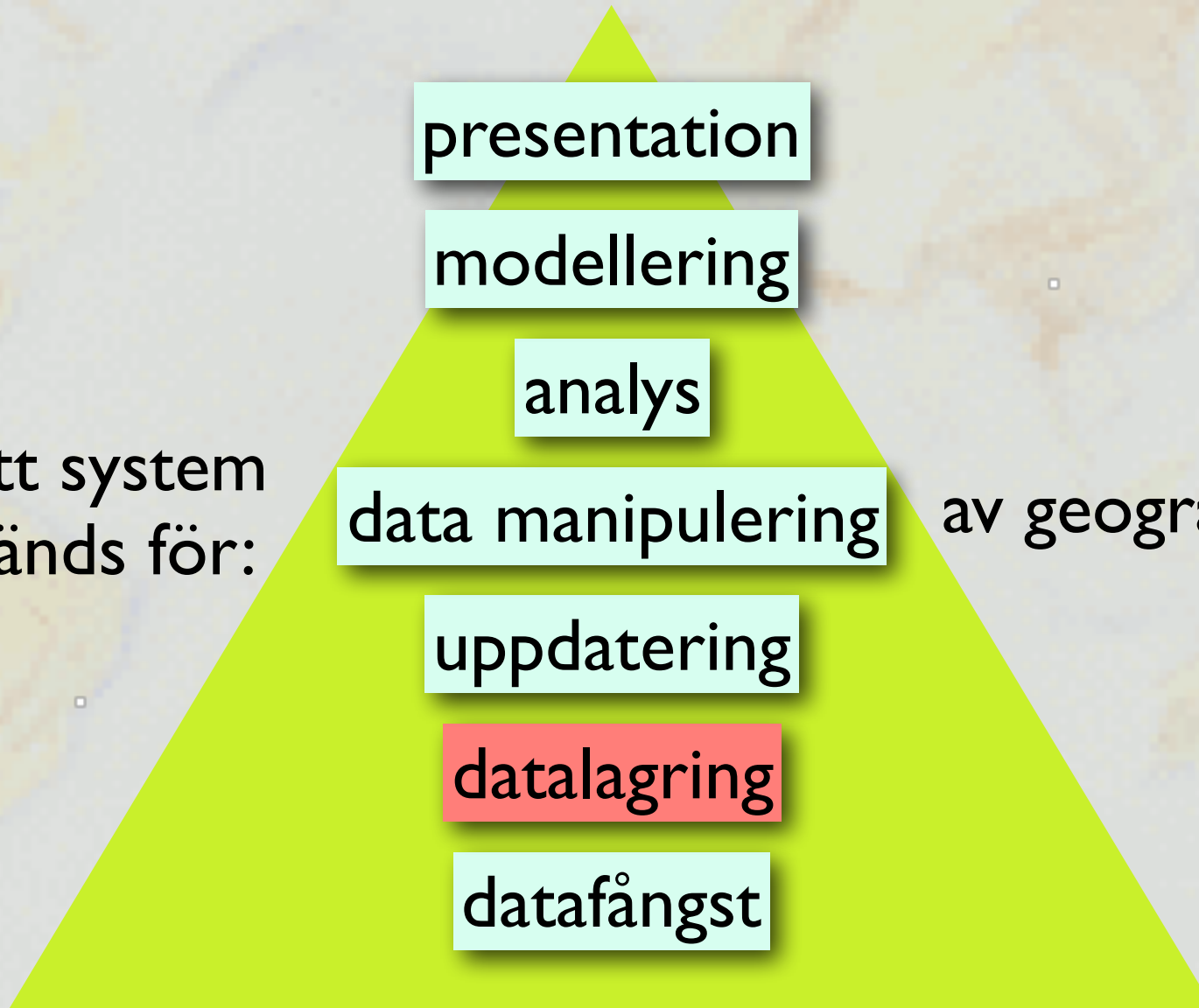
Föreläsningens innehåll och syfte

Föreläsningen ger en introduktion till datamodeller för
Geografiska Informationssystem

- Binära dataformat
- Verklighet och model
- Objektmodel och fältmodel
- Raster data strukturer
- Vektor data strukturer

Komponenter i GIS

GIS är ett system
som används för:



av geografiska data

Datastrukturer och modeller

- GIS kräver att både kartor och attributdata representeras som siffror
- Konvertering av kartor till siffror kräver en väldefinierad standard för att geografiskt kodifiera lokalisering av kartdata
- Ett koordinatsystem är en standardiserad metod för geokodning
- Standardiserade koordinatsystem använder absoluta positioner, definierade av sferoid / datum (relativa koordinatsystem - med lokalt datum vanliga)
- I ett geografiskt koordinatsystem är normalt x-riktningen öst-väst, och y-riktningen nord-syd (undantag finns)
- Vanligtvis ökar koordinatvärdena åt öster och åt norr (undantag finns)

Datastrukturer och modeller

- Digitala kartor kräver entydiga och väldefinierade begrepp, och strikt regel-baserad semantik för att kunna:
 - representera en geografisk verklighet i form av en model
 - identifiera den rumsliga utbredning av ensklida objekt
 - lokalisera objekt i ett 2D/3D koordinatsystem
 - separera intilliggande objekt från varandra
 - identifiera och sortera objekt beroende på orientering, storlek, läge etc

Datastrukturer och modeller

- En digital karta består av geografiska objekt, och attribut knutna till dessa objekt
- GIS organiserar denna geografiska data i filer och kataloger på en hårddisk
- Data kan lagras antingen som
 - binärt kodad (effektivare)
 - ASCII text (direkt läs- och editerbart)

Datastrukturer och modeller

det binära talsystemet

1 Bit



can be



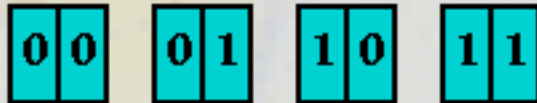
or



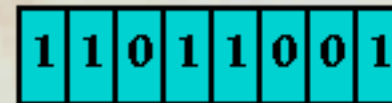
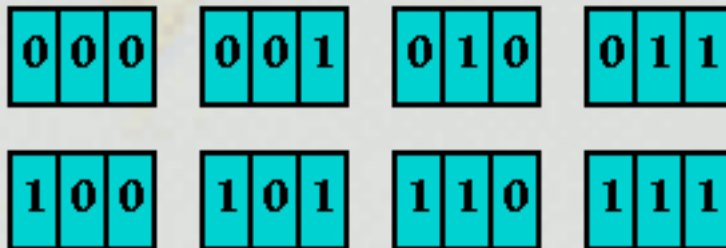
1 Byte = 8 Bits



2 Bits = 4 States



3 Bits = 8 States



1 x 2⁰ = 1 x 1 = 1
0 x 2¹ = 0 x 2 = 0
0 x 2² = 0 x 4 = 0
1 x 2³ = 1 x 8 = 8
1 x 2⁴ = 1 x 16 = 16
0 x 2⁵ = 0 x 32 = 0
1 x 2⁶ = 1 x 64 = 64
1 x 2⁷ = 1 x 128 = 128

$$1 + 8 + 16 + 64 + 128 = 217$$

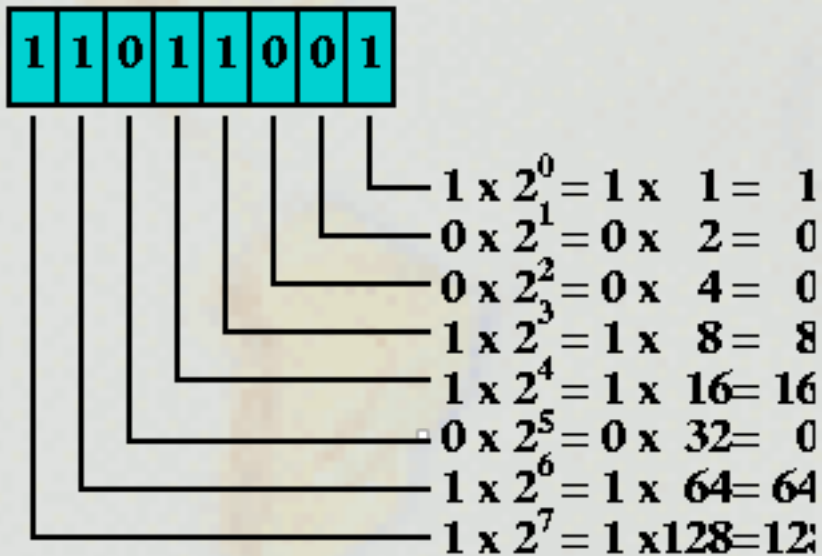
Binär talrepresentation

Benämning	Bitar	tecken	Värdeområde
Byte	8	signed	-127..127
Byte	8	unsigned	0..255
Small integer	16	signed	-32768..32768
Word	16	unsigned	0..65535
Integer	32	signed	-2147483648..2147483648
Cardinal	32	unsigned	0..4294967295
Single	32	7-8 decimaler	$1.5 \cdot 10^{-45} \dots 3.4 \cdot 10^{38}$
Real48	48	11-12 decimaler	$2.9 \cdot 10^{-39} \dots 1.7 \cdot 10^{38}$
Double	64	15-16 decimaler	$5.0 \cdot 10^{-324} \dots 1.7 \cdot 10^{308}$

Binär talrepresentation

LSB = Least Significant Bit/Byte

MSB = Most Significant Bit/Byte



$$1 + 2 + 16 + 64 + 128 = 217$$

I exemplet sitter MSB i den första positionen. Om strängen inverteras hamnar istället MSB i den sista positionen.

På samma sätt kan ett integer tal (16 bitar = 2 byte) konstrueras med MSByte i första positionen = big endian, eller med LSByte i första positionen = small endian.

Binär talrepresentation

ASCII American National Standards Institute

```
!"#$%&'()*+,-./  
0123456789:;<=>?  
@ABCDEFGHIJKLMNO  
PQRSTUVWXYZ[\]^_  
`abcdefghijklmnop  
qrstuvwxyz{|}~
```

Verklighet och modell

Världen är oändligt komplex

- Innehållet i en databas representerar en begränsad syn på verkligheten; en rumslig databas är en av oändligt många möjliga representationer av modeller av verkligheten
- Ontologiska aspekter
- Epistemological aspekter
- Användarens tillgång till och tolkning av en rumslig databas är via ett gränssnitt

Verklighet och modell

En rumslig databas kan innehålla

- Digitala abstraktioner av verkliga objekt
 - ex.v. land, vatten, hus, vägar, träd
- Digitala abstraktioner av fiktiva objekt
 - ex.v. politiska gränser, ekosystem

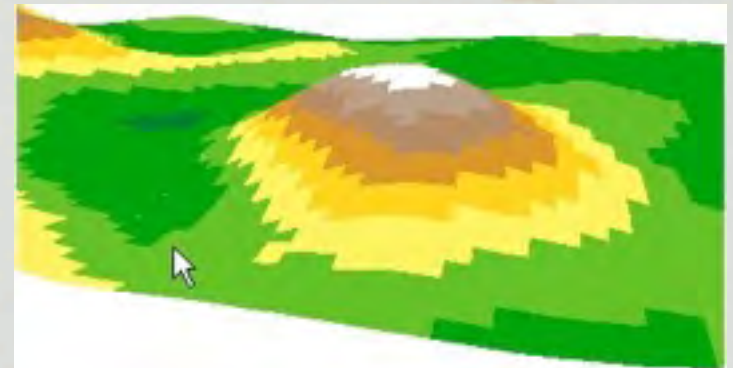
Verklighet och modell

Datorer är bra på att lagra diskret data, men sämre på att lagra kontinuerlig data - till syvende och sist är allt lagrat som 1 eller 0.

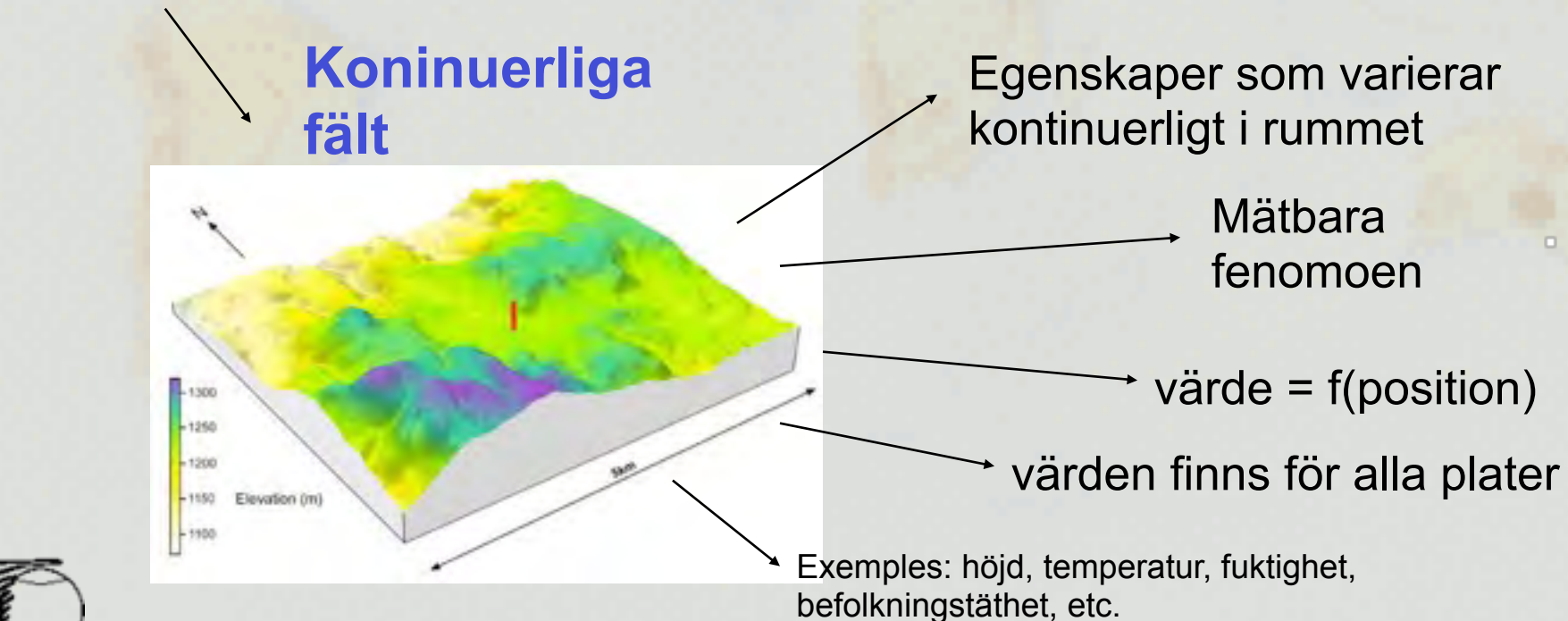
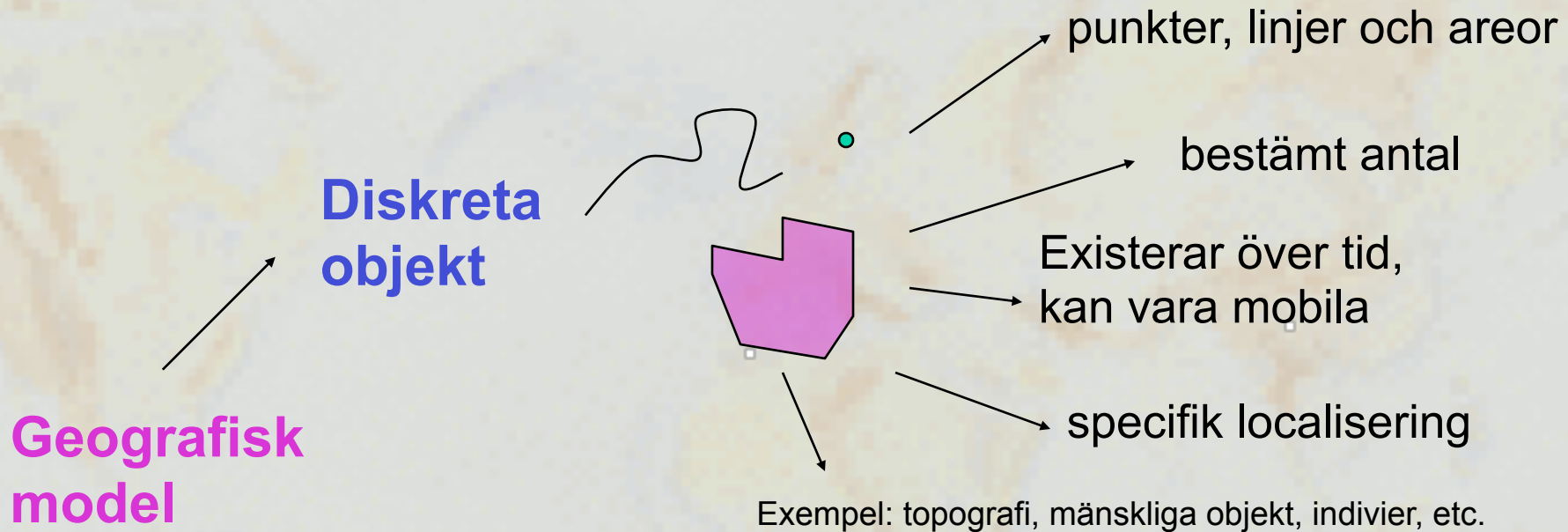


Verklighet och modell

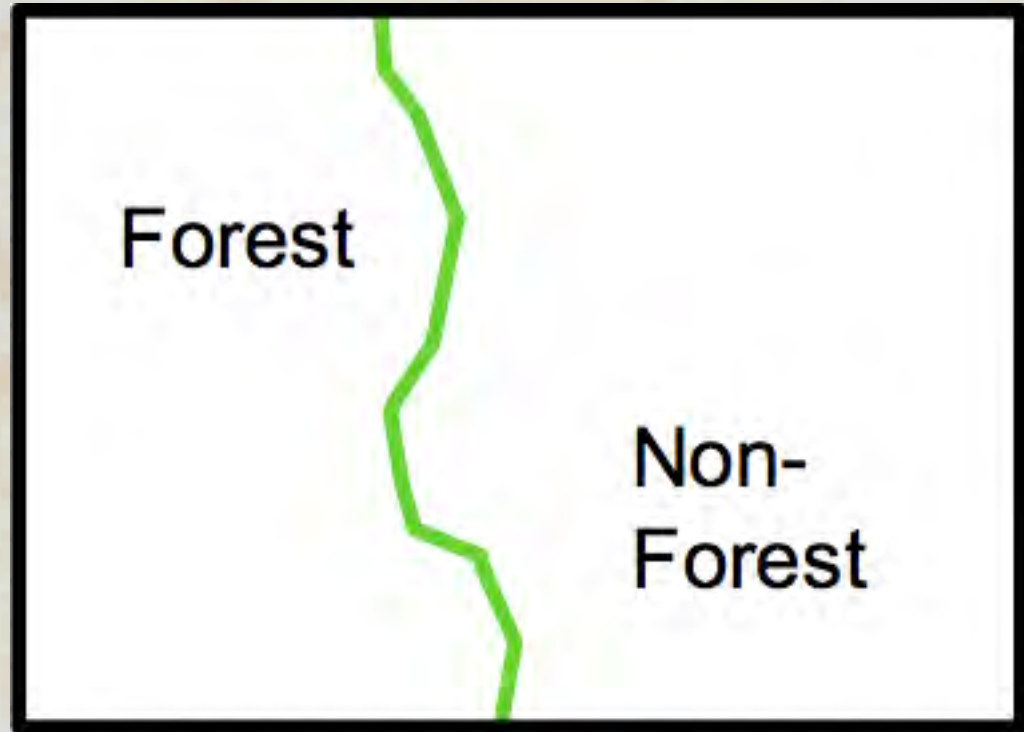
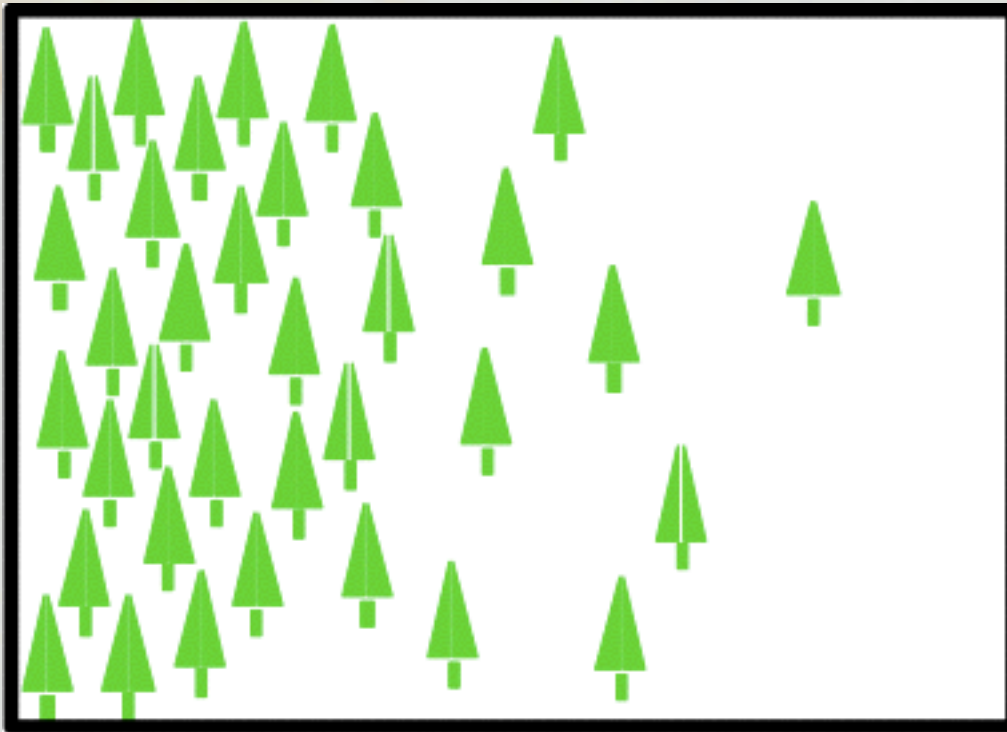
- Objekt som är av diskret natur, hus, vägar, distrikt etc, vållar inga problem att representera som diskreta objekt.
- Egenskaper som finns överallt och som varierar kontinuerligt, elevation, temperatur, lufttryck, måste approximeras till en diskret representation.



Verklighet och modell



Verklighet och modell



Diskreteringen av kontinuerliga fenomen
är ofta godtycklig

Objektmodell och fältmodell

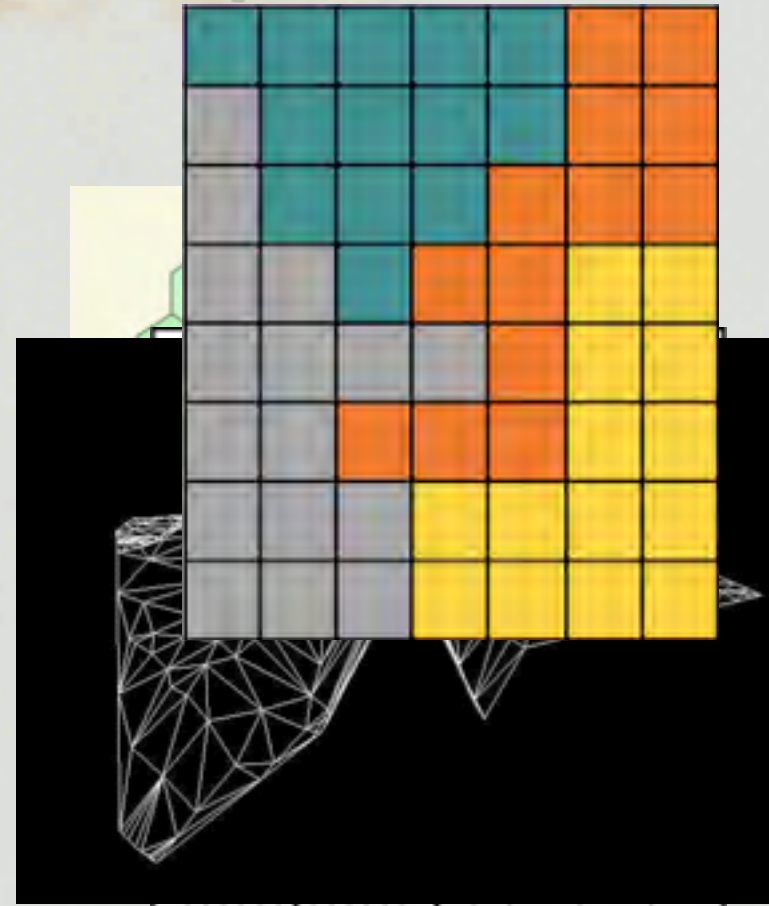
GIS-samhället har utvecklat konceptuella modeller av verkligheten, sprungna ur kartografi snarare än datalogi:

- Objektmodell - punkter, linjer, ytor fyller upp alla delar av rummet
- Fältmodell - Värderna för varje position

Tesseleringsmodeller

Raster data modellen tillhör en större grupp av fältdatamodeller eller tesseleringsmodeller:

- Grid eller raster
- Hexagonaler
- Triangular Irregular Network (TIN)
- Kvadratträd



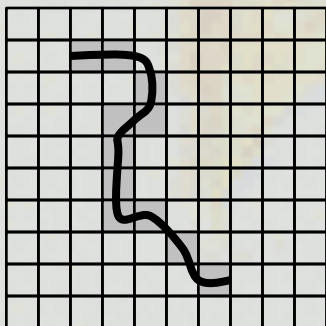
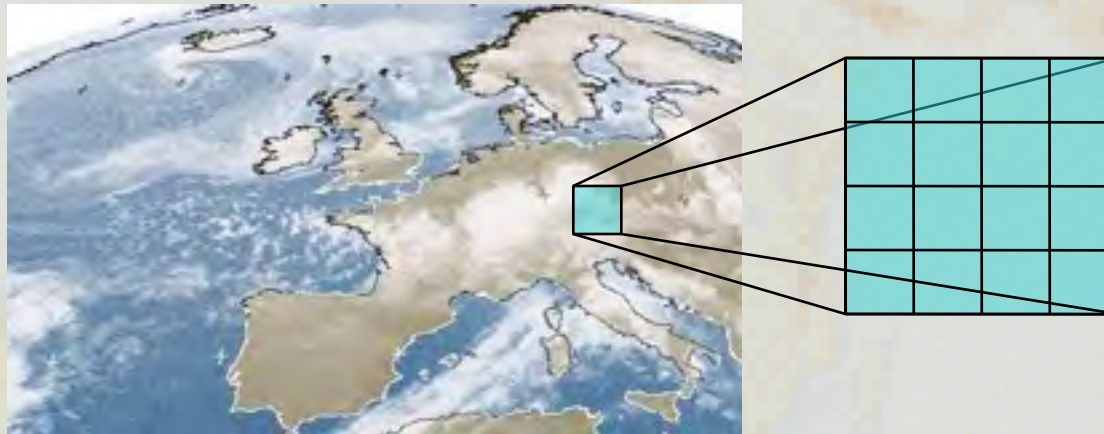
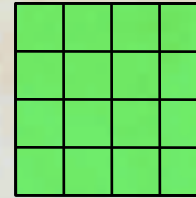
Fältmodell

Raster = regelbunden tesselering

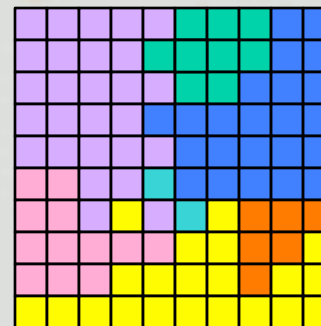
- Delar upp världen i rektangulära celler
- Registrerar grid-hörnen till en geografisk punkt
- Representerar diskreta objekt som grupper av celler med eller utan attributkoppling (koppling via indexnummer)
- Representerar fält som cellvärden (utan attributkoppling)
- Värden för varje cell
- Även celler utan relevant data lagras, som “ingen data”
- Vanligare att använda för fältobjekt
- Lätt att förstå

Raster data struktur

- Delar upp världen i rektangulära **celler = pixlar**
- Registrerar grid-hörnen till en geografisk punkt

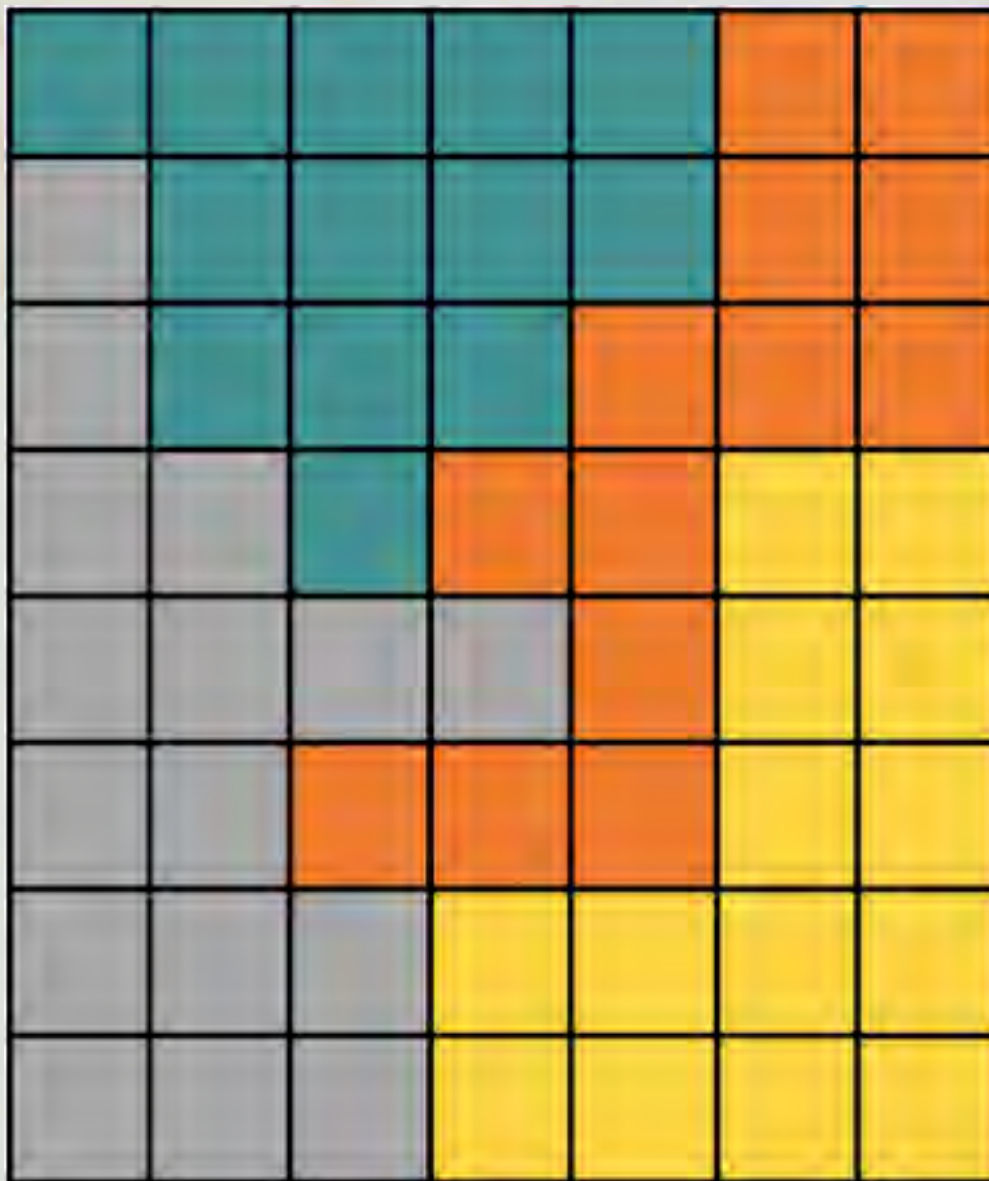


diskreta objekt
grupper av celler



Kontinuerliga fält
cellvärdet = fältvärdet

Raster data struktur

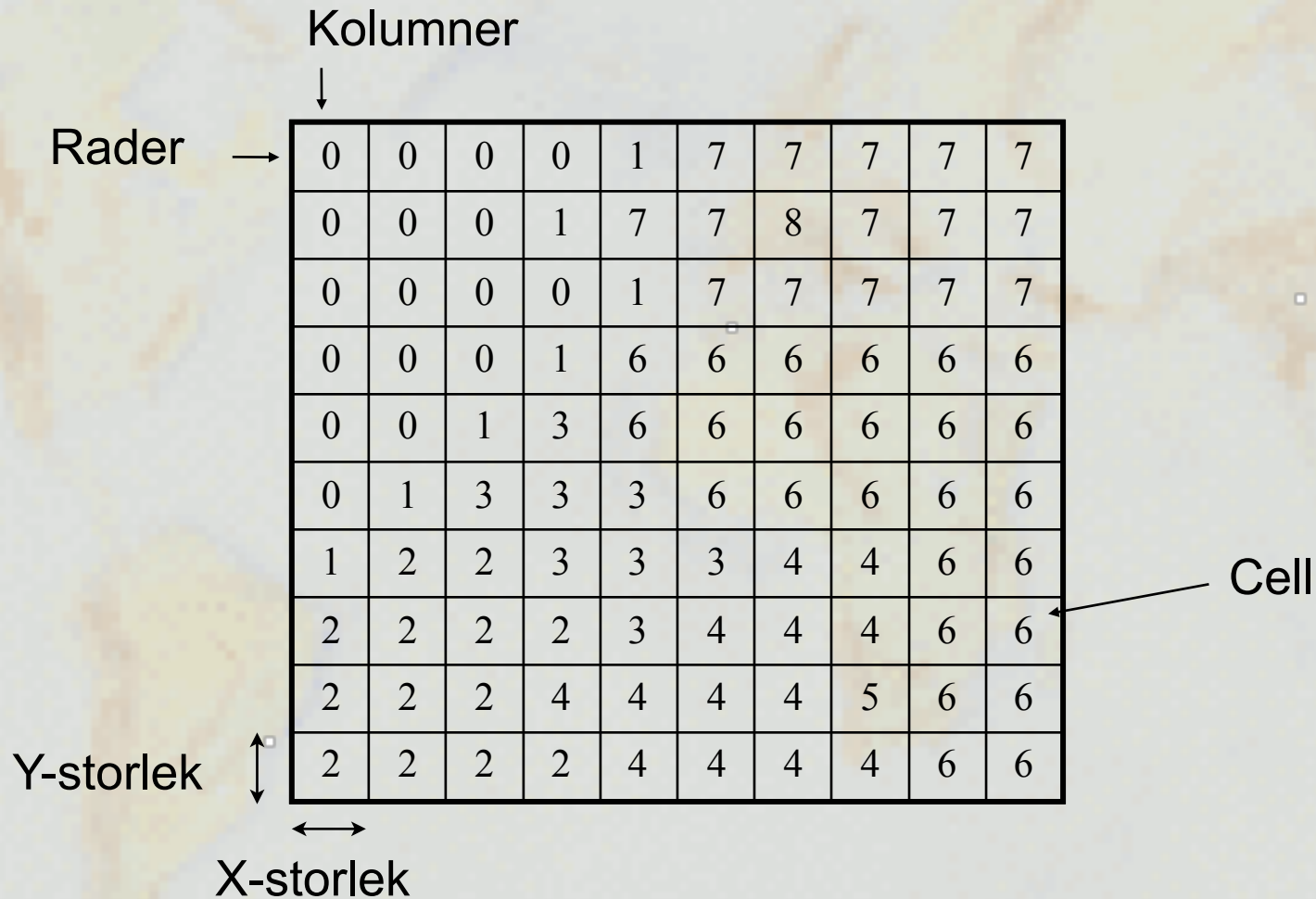


Raster data struktur
för diskreta objekt -
markanvändning.

Raster

- Pixel storlek
 - Storleken på cellen eller bildelementen som definierar den rumsliga detaljeringsgraden/ upplösningen
 - kan vara olika i x och y
- Tilldelning av cellvärden - värdet på en cell kan representera
 - medelvärdet för cellens yta
 - typvärdet för cellens yta
 - mittvärdet för cellets yta

Raster data struktur



Cell/pixel storlek = **rumslig upplösning**

- definierar detaljnivå för rumsliga objekt
- variationer inuti pixeln går förlorad

Raster

Mixade pixlar - ett problem med raster



Vatten dominant

V	V	L
V	V	L
V	V	L

Segraren tar allt

V	L	L
V	L	L
V	L	L

ekotoner som egen klass

V	E	L
V	E	L
V	E	L

Raster

Lagring av rasterdata

- Sekventiell lagring
 - Byte Interleaved by Pixel - BIP
 - Byte Interleaved by Line - BIL
 - Band Sequential - BSQ
- Blockkodning
- Kedjekodning
- Radlängdskodning
- Kvadratträd

Raster

Filstorlek

$\text{rader} * \text{kolumner} * \text{Byte per pixel} = \text{filstorlek}$

Storlek på fil med byte-värden (1 byte per pixel)

$\text{rader} * \text{kolumner}$

Raster

metadata och huvudfil

Exempel I: Byte data
ERmapper

```
DatasetHeader Begin
  Version          = "5.5"
  Description      = "NOAA-AVHRR NDVI annual average "
  DataSetType     = ERStorage
  DataType        = Raster
  ByteOrder       = LSBFirst
  CoordinateSpace Begin
    Datum          = "CLARKE 1866"
    Projection     = "ALBERSEA"
    CoordinateType = EN
    Rotation       = 0:0:0.0
  CoordinateSpace End
  RasterInfo Begin
    CellType       = Unsigned8BitInteger
    NullCellValue = 0
    CellInfo Begin
      Xdimension   = 8000
      Ydimension   = 8000
    CellInfo End
    NrOfLines     = 360
    NrOfCellsPerLine = 450
    RegistrationCoord Begin
      Eastings     = -3920000
      Northings    = 3250000
    RegistrationCoord End
    NrOfBands     = 1
    BandId Begin
      Value        = "Pseudo"
    BandId End
  RasterInfo End
DatasetHeader End
```

Raster

metadata och huvudfil

Exempel 1: Byte data
ArcGIS

```
;  
;ArcView Image Information  
; NOAA-AVHRR NDVI annual average  
; Projection: ALBERS (Albers Equal Area Conic)  
; Units: METERS  
; Spheroid: CLARKE1866  
; 1st standard parallel (dms): -19 00 0.000  
; 2nd standard parallel (dms): 21 00 0.000  
; central meridian (dms): 20 00 0.000  
; latitude of projection origin: 1 00 0.000  
; false easting (meters): 0.00000  
; false northing (meters): 0.00000  
;  
NCOLS 450  
NROWS 360  
NBANDS 1  
NBITS 8  
LAYOUT BIL  
BYTEORDER 1  
SKIPBYTES 0  
MAPUNITS METERS  
ULXMAP -3916000  
ULYMAP 3246000  
XDIM 8000.00000  
YDIM 8000.00000
```

Raster

metadata och huvudfil

Exempel I: Byte data
IDRISI

file format : IDRISI Raster A.I
file title : NOAA-AVHRR NDVI annual average
data type : byte
file type : binary
columns : 450
rows : 360
ref. system : albersaf
ref. units : m
unit dist. : 1.0000000
min. X : -3920000.0000000
max. X : -320000.0000000
min. Y : 370000.0000000
max. Y : 3250000.0000000
pos'n error : unknown
resolution : 8000.0000000
min. value : 0
max. value : 255
display min : 0
display max : 255
value units : unspecified
value error : unknown
flag value : none
flag def'n : none
legend cats : 0

Raster

metadata och huvudfil

Exempel 1: Byte data ENVI

ENVI

description = {

NOAA-AVHRR NDVI annual average }

samples = 450

lines = 360

bands = 1

header offset = 0

file type = ENVI Standard

data type = 1

interleave = bsq

sensor type = AVHRR

byte order = 0

map info = {Albers_NDVI_ADDS, 1.0000, 1.0000, -3916000, 3246000, 8.0000000000e+003, 8.0000000000e+003, , units=Meters}

projection info = {9, 6378206.4, 6356583.8, 1.000000, 20.000000, 0.0, 0.0, -19.000000, 21.000000, Albers_NDVI_ADDS, units=Meters}

wavelength units = Unknown

band names = {

NDVI}

Raster

metadata och huvudfil

Exempel 1: Byte data
DIVA

```
Version=4.1
Title=NDVlg Annual mean 2004
Created=20050306
[GeoReference]
Projection=ALBERS
Datum=CLARKE1866
Mapunits=m
Columns=450
Rows=360
MinX=-3920000
MaxX=-32000
MinY=37000
MaxY=3250000
ResolutionX=8000
ResolutionY=8000
[Data]
DataType=BYTE
MinValue=0
MaxValue=255
NoDataValue=-9999
Transparent=1
Units=NDVI
[Application]
```


Raster

metadata och huvudfil

Exempel 1: Byte data

JPG (*.jpw, *.jpgw)

TIF (*.tfw)

BMP (*.bmpw)

8000

0

0

-8000

-3916000

3246000



Denna typ av huvudfil kallas
“world”-fil och kan följa med
alla typer av bildformat.

Raster

metadata och huvudfil

Exempel 2: Integer data
ERmapper

```
DatasetHeader Begin
  Version          = "5.5"
  Description      = "NOAA-AVHRR NDVI annual npp "
  DataSetType     = ERStorage
  DataType        = Raster
  ByteOrder       = LSBFirst
  CoordinateSpace Begin
    Datum          = "CLARKE 1866"
    Projection     = "ALBERSEA"
    CoordinateType = EN
    Rotation       = 0:0:0.0
  CoordinateSpace End
  RasterInfo Begin
    CellType       = Unsigned16BitInteger
    NullCellValue = 0
    CellInfo Begin
      Xdimension   = 8000
      Ydimension   = 8000
    CellInfo End
    NrOfLines     = 360
    NrOfCellsPerLine = 450
    RegistrationCoord Begin
      Eastings     = -3920000
      Northings    = 3250000
    RegistrationCoord End
    NrOfBands     = 1
    BandId Begin
      Value        = "Pseudo"
    BandId End
  RasterInfo End
DatasetHeader End
```

Raster

metadata och huvudfil

Exempel 3: real data
ERmapper

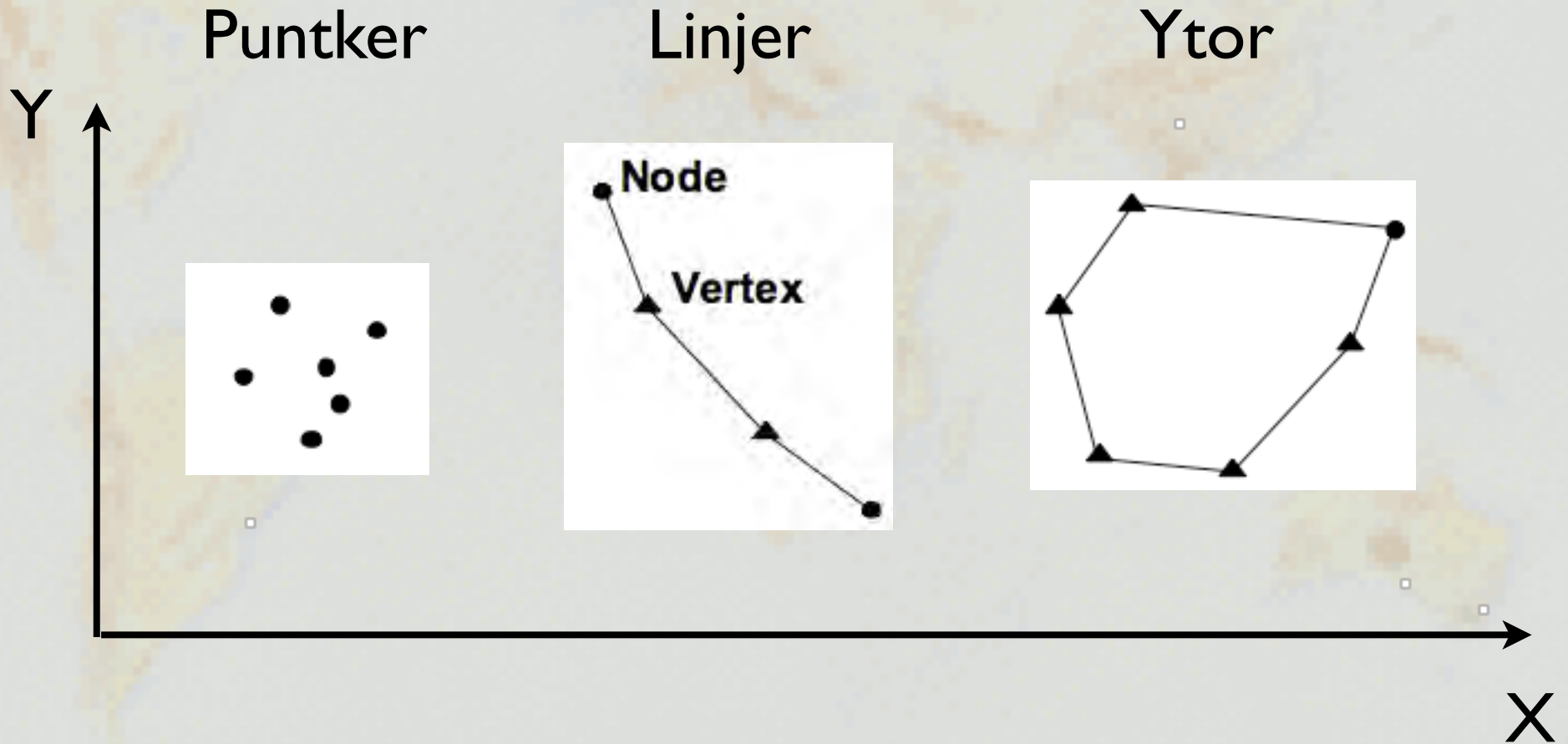
```
DatasetHeader Begin
  Version          = "5.5"
  Description      = "NDVI annual max trend 1982-2004"
  DataSetType     = ERStorage
  DataType        = Raster
  ByteOrder       = LSBFirst
  CoordinateSpace Begin
    Datum          = "CLARKE 1866"
    Projection     = "ALBERSEA"
    CoordinateType = EN
    Rotation       = 0:0:0.0
  CoordinateSpace End
  RasterInfo Begin
    CellType       = IEEE32REAL
    NullCellValue = 0
    CellInfo Begin
      Xdimension   = 8000
      Ydimension   = 8000
    CellInfo End
    NrOfLines     = 360
    NrOfCellsPerLine = 450
    RegistrationCoord Begin
      Eastings     = -3920000
      Northings    = 3250000
    RegistrationCoord End
    NrOfBands     = 1
    BandId Begin
      Value        = "Pseudo"
    BandId End
  RasterInfo End
DatasetHeader End
```

Vektor data model

Verkliga eller fiktiva objekt representerade som punkter, linjer och ytor

- punkter representerar objekt utan utbredning, eller med för skalan irrelevant utbredning
- linjer knyter samman punkter till start-, bryt-, och stoppunkter
- ytor (polygoner) byggs upp av slutna linjer

Vektor data model



Vektor data model

Precision och noggrannhet

- Objekt definieras av x,y koordinater relaterade till ett koordinatsystem (long/lat eller x,y).
- Precision (upplösning) i koordinater beror på binär lagringsform (6-15 decimaler), men är ofta hög
- Noggrannheten i data oftast mer begränsande än upplösning

Vektor data model

Precision och noggrannhet

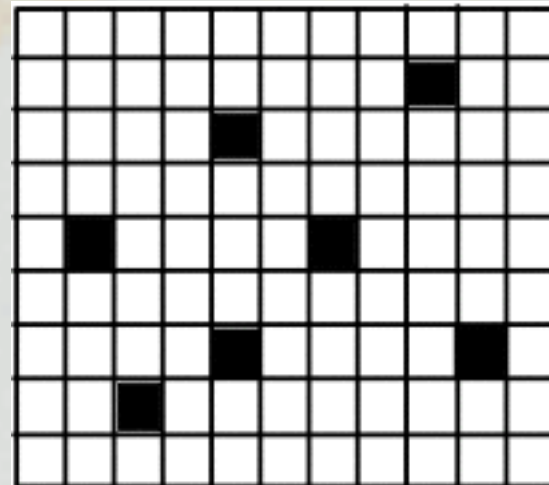
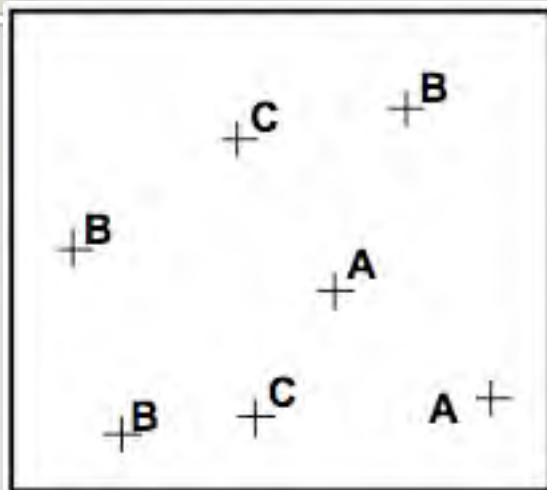
- Precision är det minsta avstånd mellan två intilliggande objekt som uppmätts och lagrats.
- Noggrannhet är frånvaro av fel
- Osäkerhet är ett mer generellt begrepp, och inkluderar både precision och noggrannhet.

Vektor data model

Punktdata

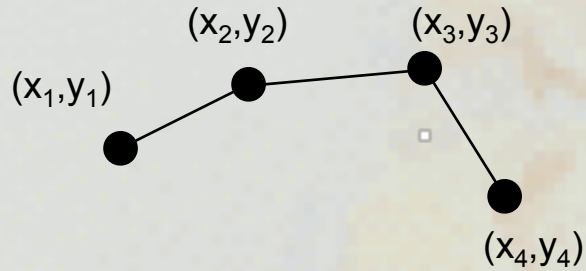
● (x,y)

Flaggstång
Byggnad
Stad

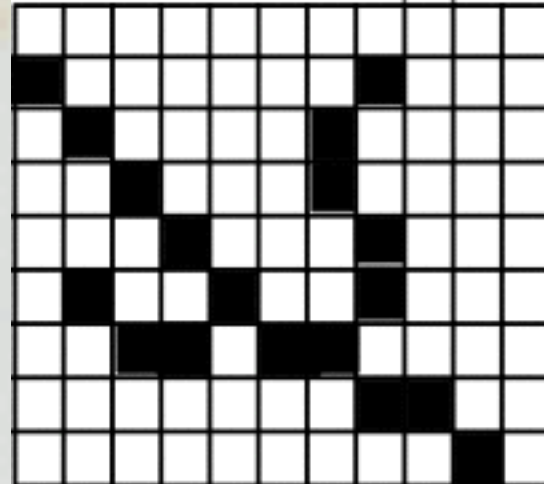


Vektor data model

Linjedata

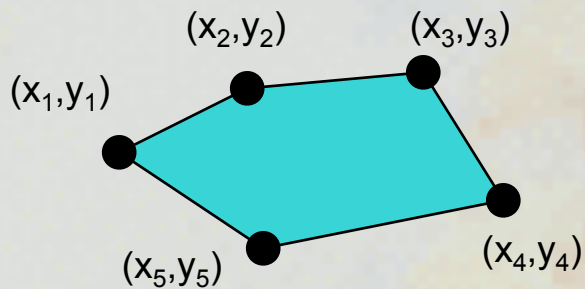


Vattendrag
Väg
Järnväg
Staket

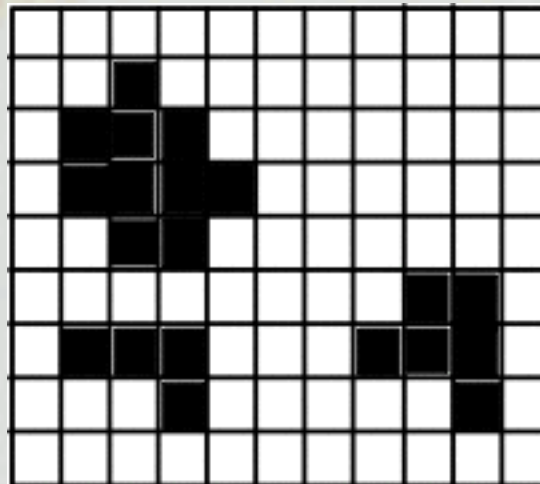
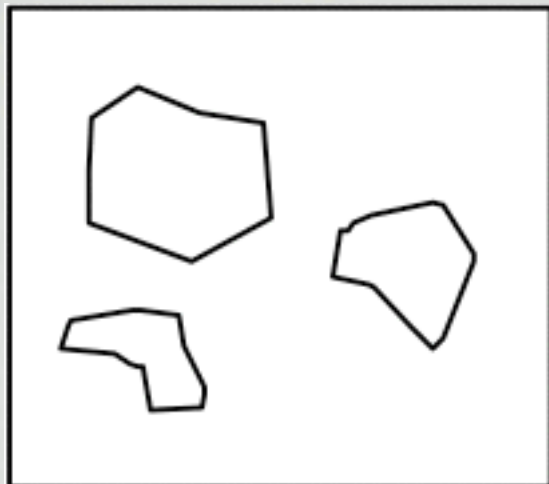


Vektor data model

Areadata



Sjö
Skog
Stad
Fastighet



Vektor data model

Tre huvudsakliga modeller för att lagra vektorer

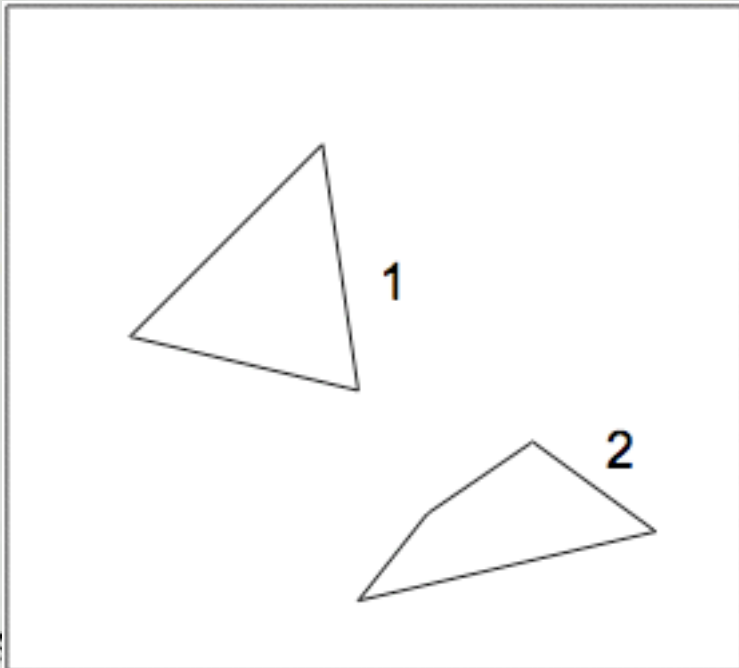
- Enkel (eller spaghetti) data struktur
 - Ingen logik, dubblering av data (inom ett lager)
- Punkt listor
 - Ingen logik, ingen dubblering
- topologisk struktur
 - Logik, ingen dubblering

Vektor data model

Spaghetti vektor data model

Varje punkt, linje eller polygon lagras i en post (“record”) som innehåller Id och koordinater som definierar geometri (de första GIS-programmen hade spaghetti data struktur)

Polygoner



ID	Coordinates
1	(2,4), (4,3), (3,6) , (2,4),
2	(3,1), (5,2), (4,3), (3,2), (3,1)

Vektor data model

Spaghetti vektor data model

■ Fördelar

- enkelt

- effektiv för display och utskrift

■ Nackdelar

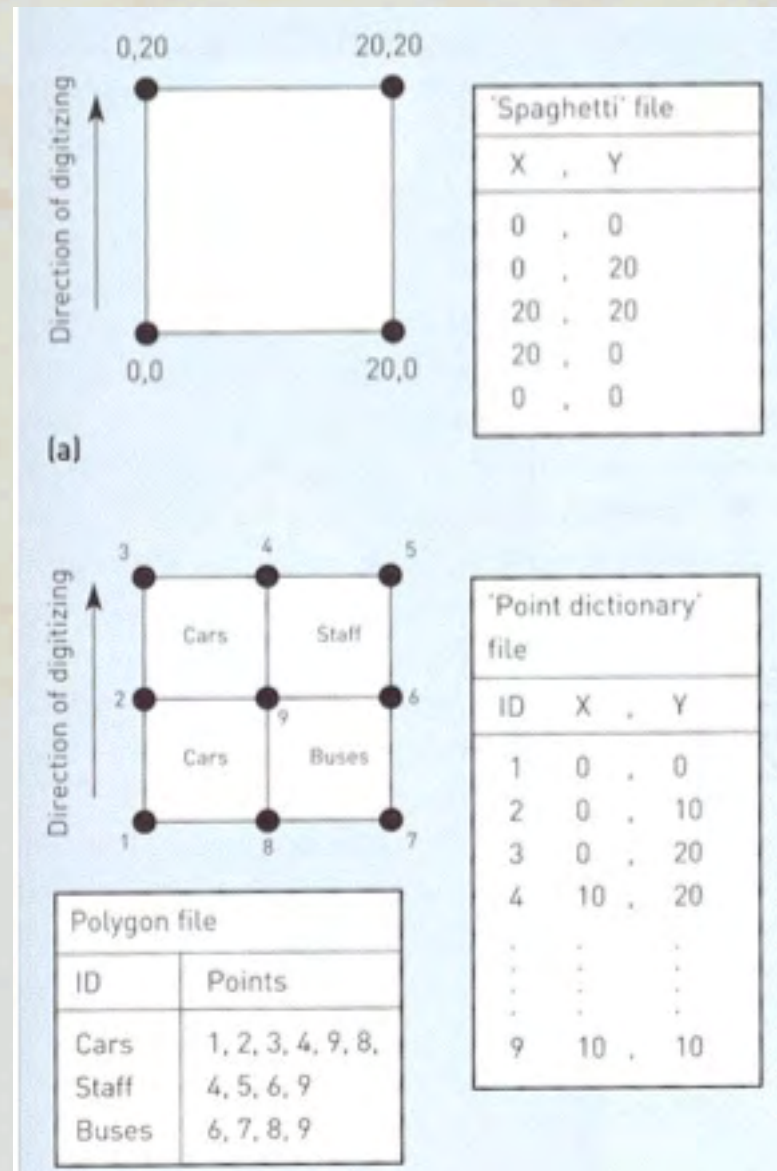
- Ineffektivt för rumsliga analyser

- och generaliseringar

Vektor data model

Punkt data struktur

Ingen data redundans
Ingen topologi



Vektor data model

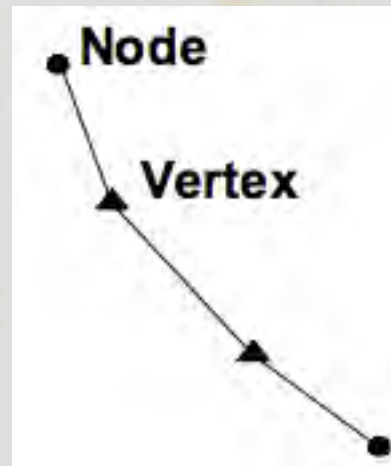
Topologisk data struktur

Nätverkstopologi

kallas även “ark-nod” modellen

ark = linje

nod = slutpunkt på en linje, eller en punkt där en linje splittras eller linjer går samman



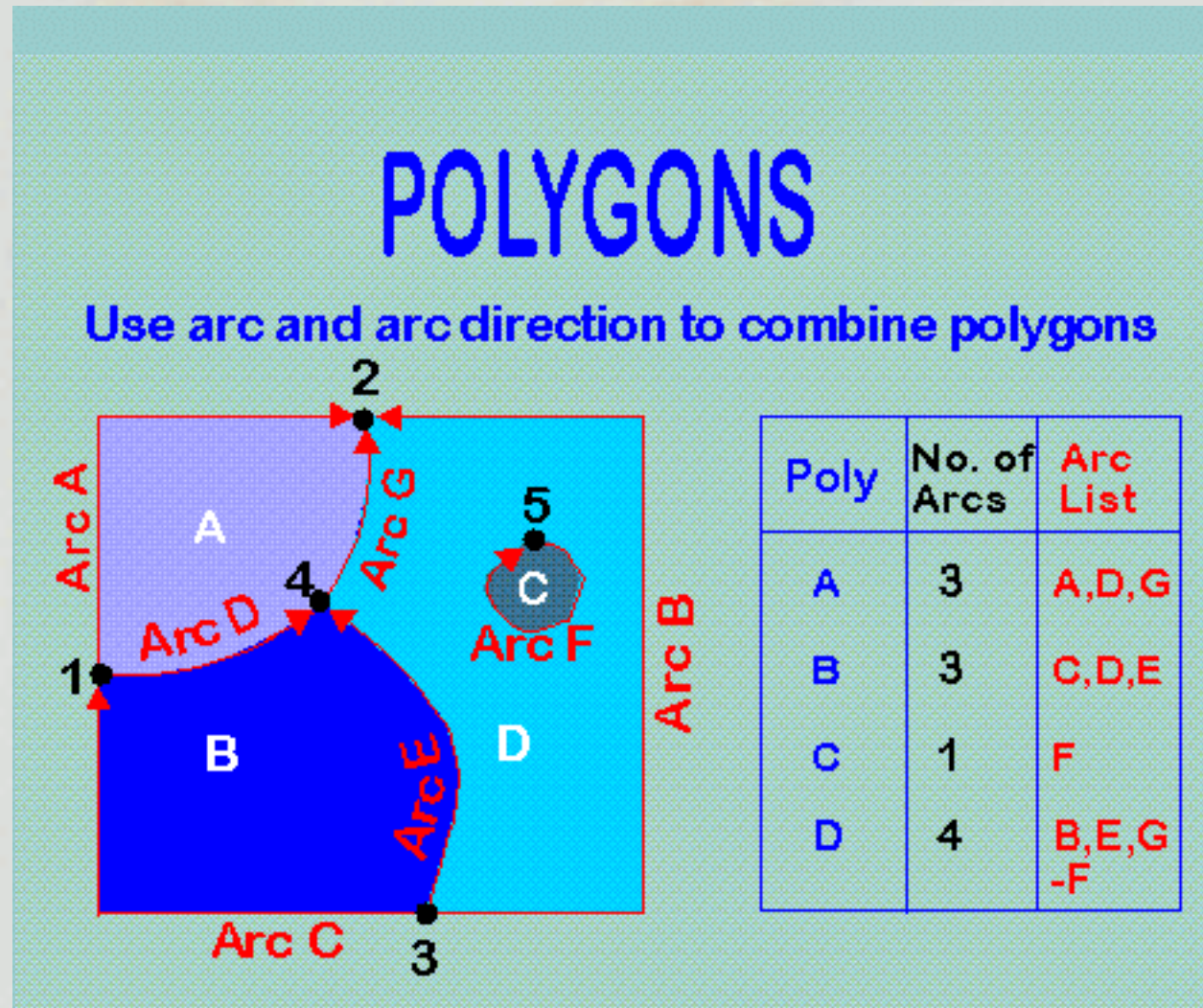
Vektor data model

Topologisk data struktur

- registrerar x/y koordinater av rumsliga objekt

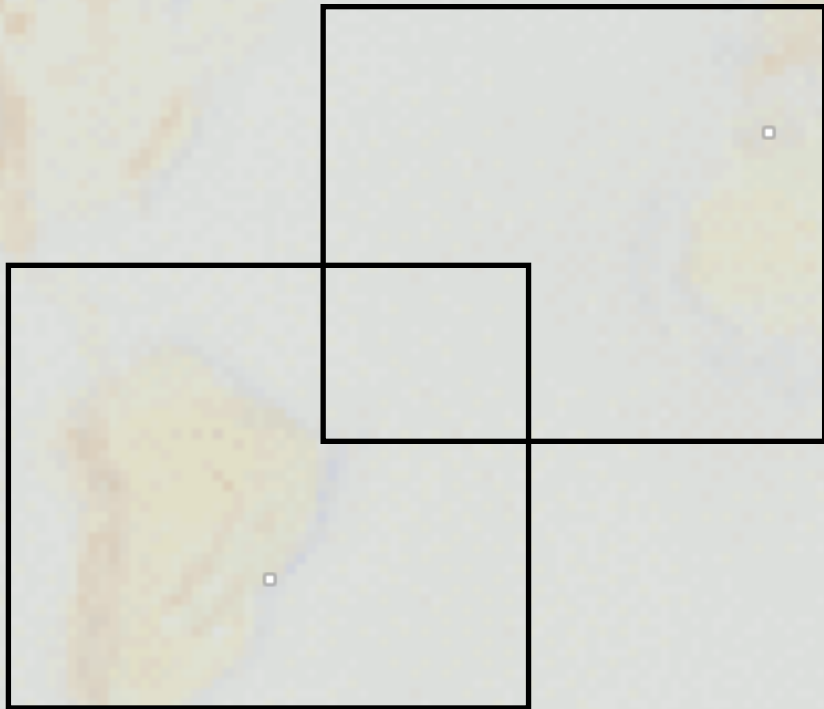
Kodar rumsliga relationer:

- vilka arkar kopplar till vilken nod
- vilka ytor ligger på sidorna av en ark
- vilka arkar bygger en polygon



Vektor data model

Spaghetti modell och topologisk modell



Spaghetti: registrering
som 2 eller 3 ytor

Topologiskt: registrering
som 3 ytor

Vektor data model

topologisk vektor data model

- **Fördelar**
 - Rumsliga relationer är explicita
 - Rumslig analys utan koordinater möjlig
- **Nackdelar**
 - komplex data struktur
 - topologi måste omregistreras efter varje uppdatering

Fördelaktigaste systemet för flertalet användare

Jämförelse mellan raster och vektor

Fördelar

- Enkel och läsbar lagring.
- Enkelt att analysera (algoritmer från fjärranalys och bildbehandling)
Enkla att kombinera (överläggning).

Raster

Nackdelar

- Kvalitet beror på pixel-storlek.
- Kräver mycket fysisk lagringskapacitet: grid formatet växer kvadratisk när cellstroleken minskar.

Fördelar

- Enkelt att skala om, kvalitet behålls vid transformationer.
- Enkelt med topologiska och nätverksberäkningar.
- Effektivt utnyttjande av fysisk lagringskapacitet.

Vektor

Nackdelar

- Beräkningsmässigt mer krävande för flera standardberäkningar (Filtrering, överläggning).

Jämförelse mellan raster och vektor

Fördelar

- Enkel och läsbar lagring.
- Enkelt att analysera (algoritmer från fjärranalys och bildbehandling)
Enkla att kombinera (överläggning).

Raster

Nackdelar

- Kvalitet beror på pixel-storlek.
- Kräver mycket fysisk lagringskapacitet: grid formatet växer kvadratisk när cellstroleken minskar.

Fördelar

- Enkelt att skala om, kvalitet behålls vid transformationer.
- Enkelt med topologiska och nätverksberäkningar.
- Effektivt utnyttjande av fysisk lagringskapacitet.

Vektor

Nackdelar

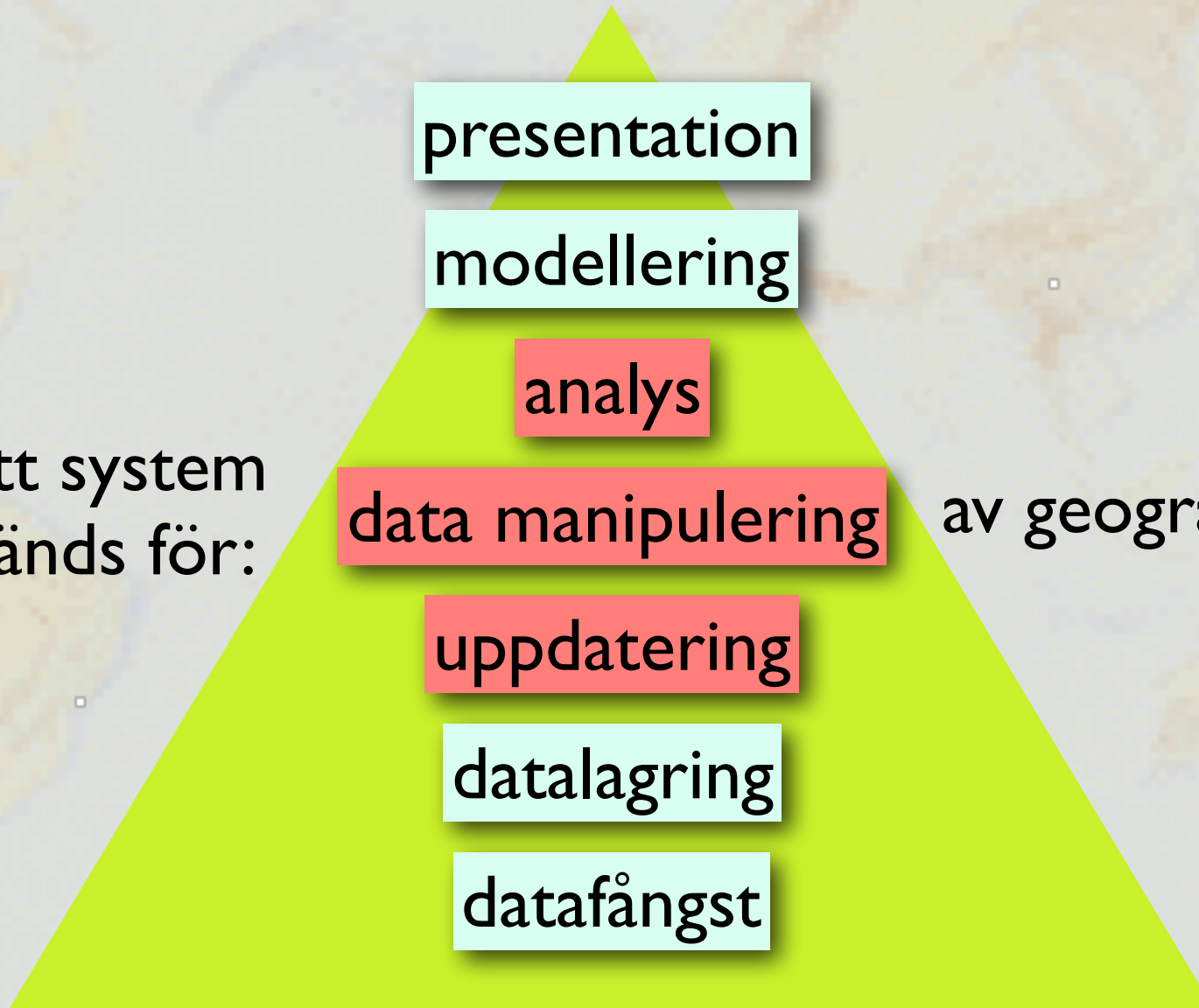
- Beräkningsmässigt mer krävande för flera standardberäkningar (Filtrering, överläggning).

Introduktion till databaser för Geografiska Informationssystem

- Databaser
- Databasutveckling
- Entity-relationship-modellen (ER)
- Konkret exempel

Komponenter i GIS

GIS är ett system
som används för:



av geografiska data

Databaser

- Databastekniken utvecklades på 1970 talet för flygbokningar etc
- Konceptuella metoder uppstod ungefär samtidigt, av vilka ER-modellen (Entity-Relationship-modellen) fortfarande används
- Objektmodellering (UML) är en modernare konceptuell metod, som liknar objekt-orienterad programmering med hierarkiska klasser och ärvda egenskaper.
- Den vanligaste formen av databas är RelationsDataBaser (RDB); när en konceptuell modell är klar översätts den till en RDB
- Vanliga RelationsDataBaser inkluderar Access, DBase, Oracle, My SQL
- De flesta RDB har anammat samma standard för hur man ställer frågor - Standar Query Language (SQL)

Databaser

Databasutvecklingsprocessen

- Samla information
 - Vilka data ska med,
 - vad ska man använda data till,
 - vem ska kunna bearbeta data etc
- Ta fram en begreppsmodell
 - Formalisera ett databasschema
 - ERmodell
 - Objektmodell (UML)
- Anpassa databasshemat till relationsdatabassystem
- Skapa databasen i relationsdatabassystem

Databaser

Entity-relationship-modellen

- Entiteter = logiska klasser hörande till databasen
- Samband = relationer mellan entiteter
- Attribut = datatyper som hör till entiteten

Databaser

Erfarenhetsregler

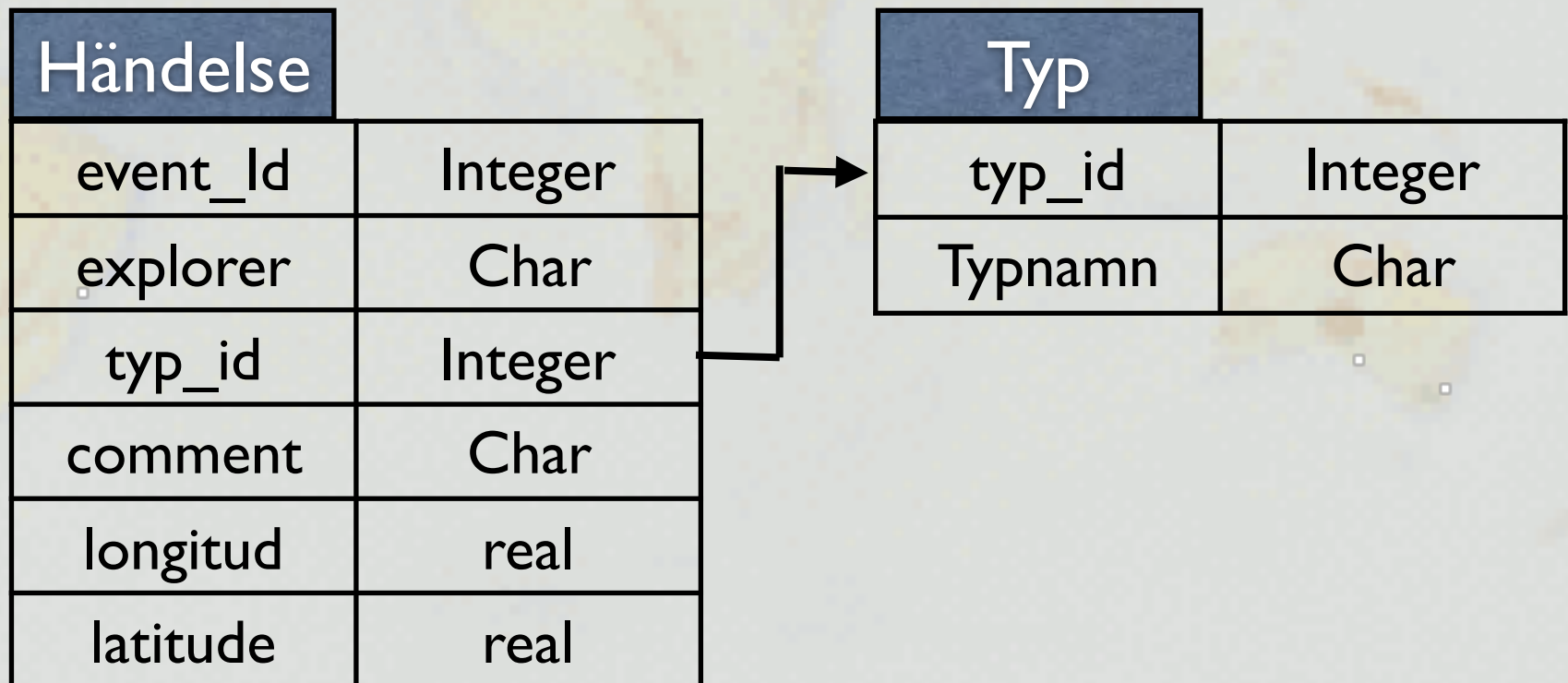
- 1. Lagra data i tabeller, där varje fält (kolumn) ska ha ett unikt namn och en entydig datatyp
- 2. Varje post (rad) i en tabell måste vara unik
- 3. Lägg fält vars värden förekommer i flera poster (rader) i tabellen i egna tabeller
- 4. Inga fält (kolumner) i tabellen får vara sammansatta av flera logiskt oberoende storheter
- 5. Inga fält (kolumner) i tabellen får innehålla upprepade värden av samma storhet

Bunta ihop reglerna 2,4 och 5 = första normalformen (1NF)

First Normal Form -> Second Normal Form -> Third Normal Form ->
-> Boyce-Codd Normal Form -> Fourth Normal Form ->
-> Fifth Normal Form -> Domain/Key Normal Form

Databaser

mapjourney punkthändelser - ett exempel



Databaser

mapjourney punkthändelser - ett exempel

