ifp

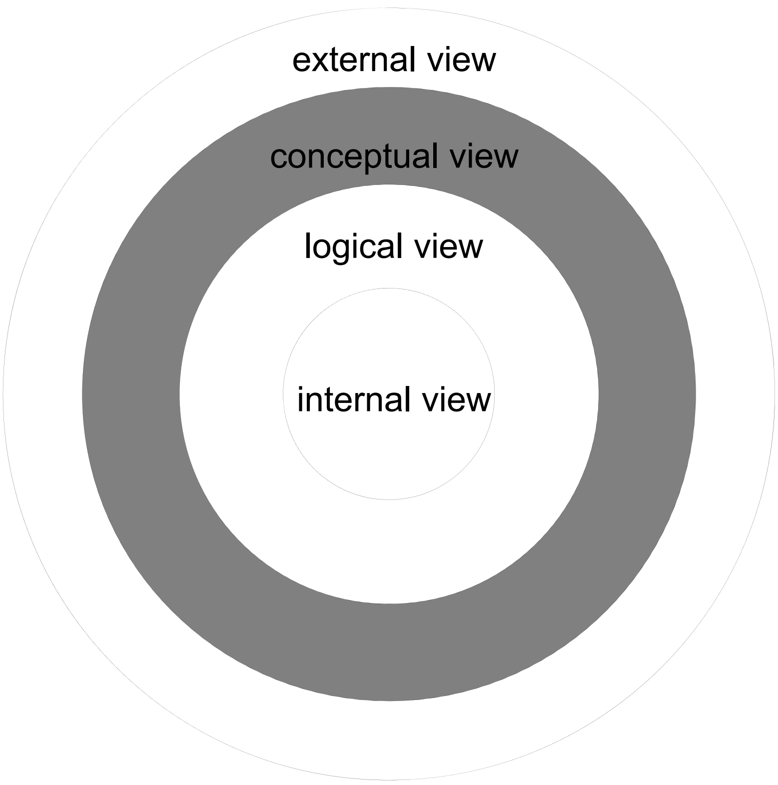
Universität Stuttgart

27.11.2019

ifp

Universität Stuttgart

Four-Layer-Model of a GIS

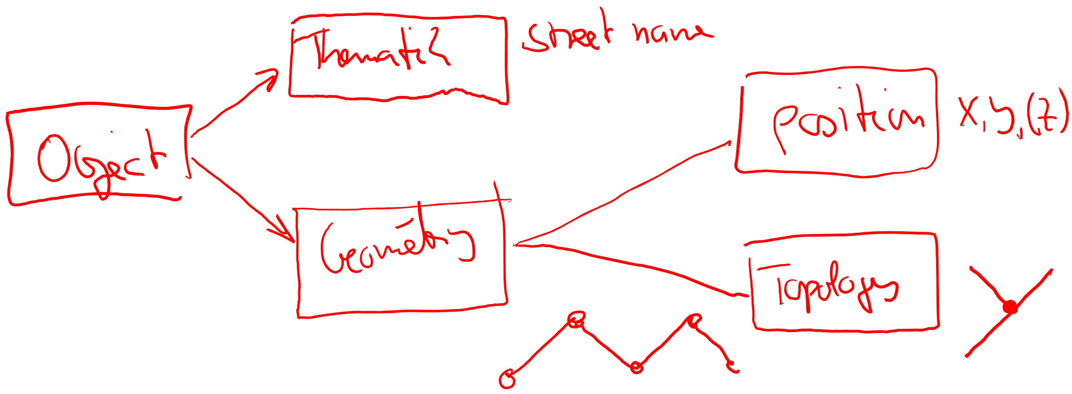


ifp

Universität Stuttgart

Conceptual Model

- conceptual model describes the transformation of the objects of the external model into (computer readable) data structures
- objects are structured according to:
  - geometry
  - topology
  - semantics (thematic information)

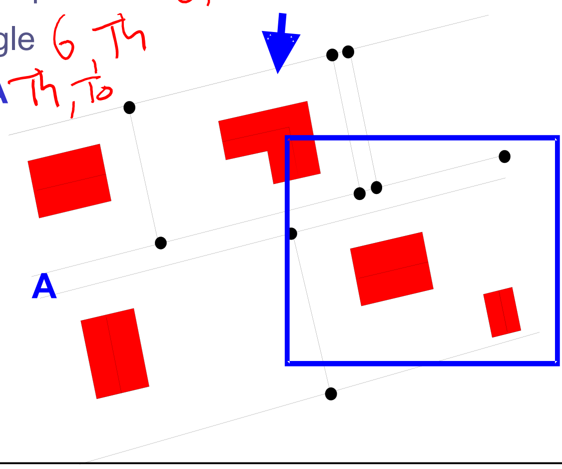


ifp

Universität Stuttgart

Queries to Spatial Information System

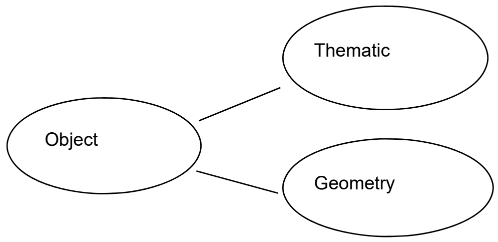
- queries concerning geometry
- queries concerning thematic information
- queries concerning topologic relations
- e.g.:
  - give identification of the line the cursor points at 6
  - give attributes of the area the cursor points at 6, Th
  - give all houses in the given rectangle 6, Th
  - give all parcels adjacent to street **A** Th, To
  - who is owner of house 121? Th
  - ...



ifp

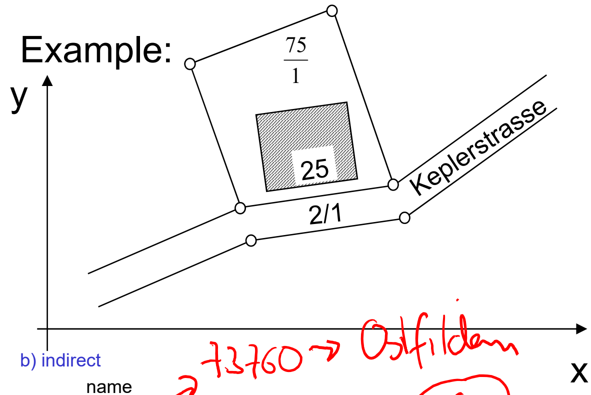
Spatial context of geodata

- The spatial context can be different:
  - exact description P = P(x,y) or P = P(x,y,z) → GIS
  - fuzzy description -> indirect → Human



a) exact (direct)

$$\Re^1 : P(x)$$
$$\Re^2 : P(x, y)$$
$$\vdots$$
$$\Re^n : P(x_1, x_2, \cdots, x_n)$$



b) indirect

- name
- zip code → 73760 → Ostfildern
- orientation (north, south, east, west)
- :

S
ES
07

Universität Stuttgart

ifp

Metric

X... coordinates

In order to analyse geometric queries, a metric must be defined.

Definition: a distance function d: X x X ⇒ $\Re_0^+$ is a metric if the following three conditions are fulfilled for all (P,Q)∈ X x X :

1) d(P,Q) is exactly than zero if P = Q

$\forall P,Q \in X \quad d(P,Q) = 0 \iff P = Q$

2) d ~~ist~~ is symmetrical

$\forall P,Q \in X \quad d(P,Q) = d(Q,P)$

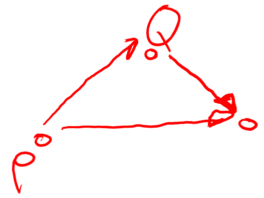
Universität Stuttgart

ifp

Universität Stuttgart

3) Triangle Inequation

$P, Q, R \in \mathbb{R}$ $\quad d(P,R) \le d(P,Q) + d(Q,R)$



Euclidian Distance

$$d(Q,P) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$$

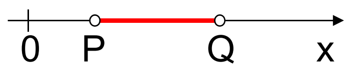
ifp

Universität Stuttgart

Metric

Exercise: Define the Euclidian metrics for $\Re^1$, $\Re^3$ and $\Re^n$

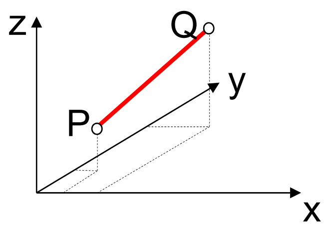
$\Re^1$:



$d(P,Q) = |x_q - x_p|$

$$\sqrt{(x_p - x_q)^2 + (y_p - y_q)^2 + (z_p - z_q)^2}$$

$\Re^3$:



$P := p(x_p, y_p, z_p)$

$Q := q(x_q, y_q, z_q)$

$d(P,Q) = ?$

$$\sqrt{\sum_{i=1}^{n} (x_{ip} - x_{iq})^2}$$

$\Re^n$:

$P := p(x_{1p}, x_{2p}, \ldots, x_{np})$

$Q := q(x_{1q}, x_{2q}, \ldots, x_{nq})$

$d(P,Q) = ?$

ifp

Raster and Vector Data

- Different types of GIS
  - raster GIS
    - which stores and analyses raster data

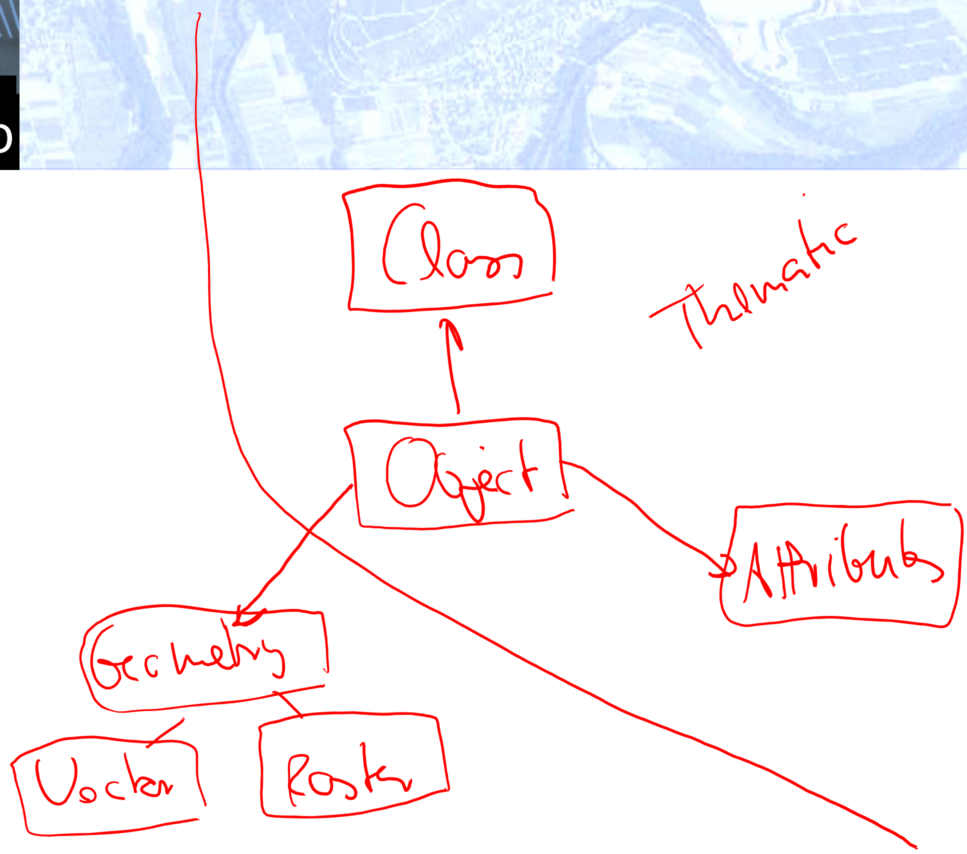
    ArcGIS Spatial Analyst
  - vector GIS
    - which stores and analyses vector data

    ArcMap
  - hybrid GIS
    - which stores and analyses raster and vector data

    Special system
    → in research
  - attention: raster data are very often used only as background information

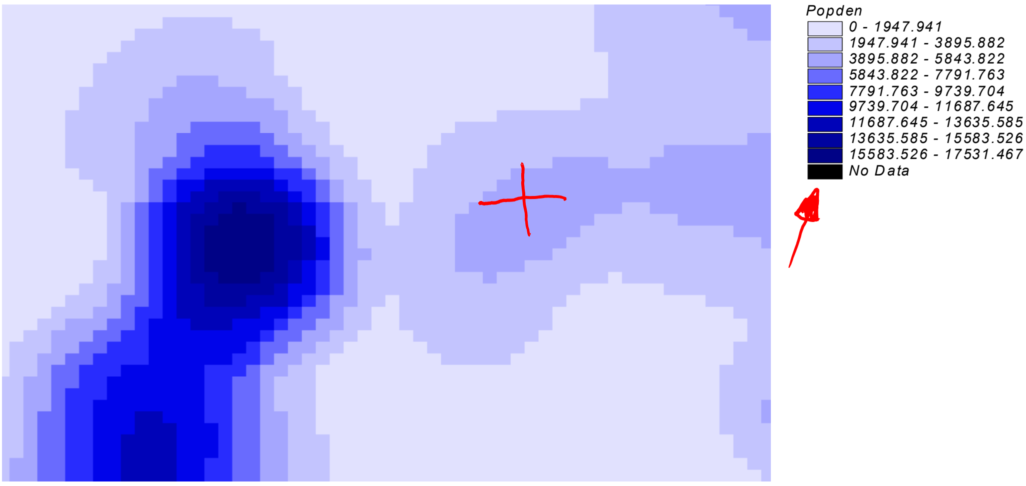
Universität Stuttgart

ifp



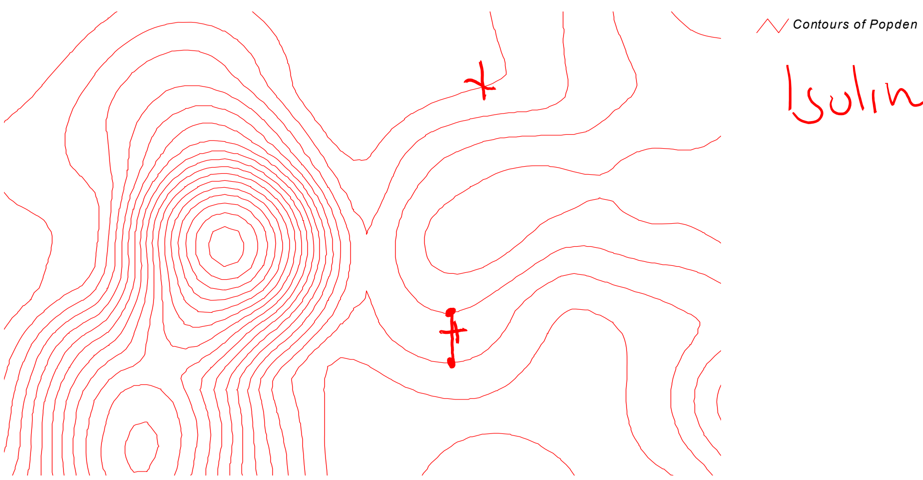
Universität Stuttgart

Example: Raster Data



Universität Stuttgart

Example: Vektor Data

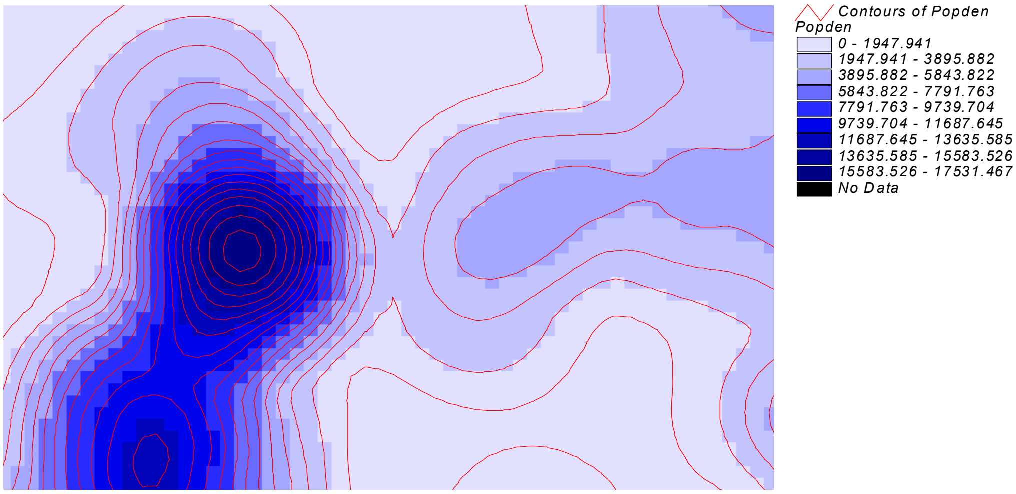


Isolines

Universität Stuttgart

ifp

Example: Hybrid Data

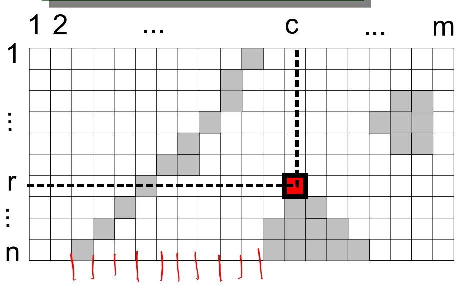


Universität Stuttgart

ifp

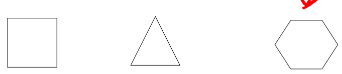
Raster Data vs. Vector Data

RASTER

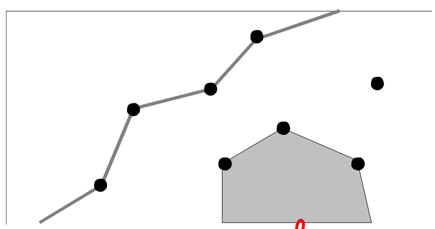


- space regularly subdivided into cells = decomposition of space
- basic unit: cell with homogenous contents
- position: discrete
- shape: square or other

medicine application

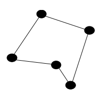


VECTOR



polygones

- set of points, lines and areas
  => construction of spatial objects
- basic units: point, line, area
- position: continuous space
  x, y, (z) coordinates
- shape: flexible



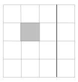
Universität Stuttgart

Raster Data vs. Vector Data

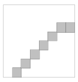
RASTER

geometrical primitives

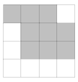
point: single cell



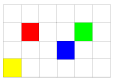
line: several cells with the same value forming a linear grouping



area: agglomeration of neighbouring cells having same value



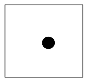
graphical description
colour



VECTOR

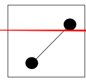
- geometrical primitives

point:



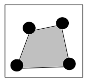
x, y

line:



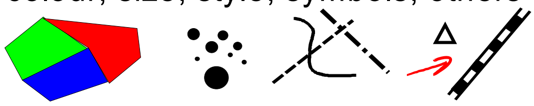
start point
(intermediate points)
end point

area:



lines

- graphical description
colour, size, style, symbols, others



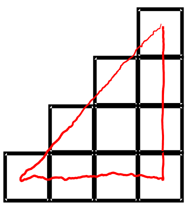
ifp
Universität Stuttgart

Raster Data vs. Vector Data

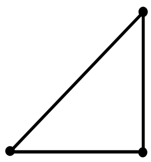
ADVANTAGES

DISADVANTAGES

RASTER	VECTOR
simple data structure (array) but large amount of data → data compression techniques: run length encoding, quadtree, ...	complex but compact data structure
positional precision depends on resolution (= minimum mapping unit) linear increase of precision (higher resolution) = increase of amount of data squared	precise coordinate positions through continuous coordinate space



imprecise
area or length
calculation

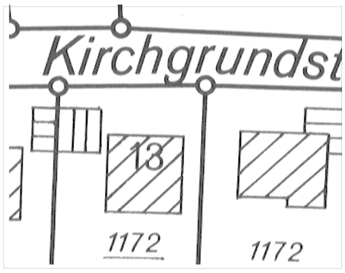


precise area
or length
calculation

Raster Data vs. Vector Data

RASTER | ADVANTAGES / DISADVANTAGES | **VECTOR**

RASTER	VECTOR
graphical representation is less aesthical (blocky appearance)	accurate graphical representation



RASTER	VECTOR
weak topology only relationship between neighbouring cells	complete encoding of topology is possible
topological analysis is difficult	efficient analysis of topological relationships (e.g. network analysis)

ifp
Universität Stuttgart

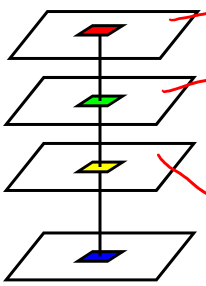
Raster Data vs. Vector Data

RASTER | ADVANTAGES / DISADVANTAGES | **VECTOR**

RASTER

- cell ≠ real world object
  individual cell does not correspond to the real world object it spatially represents (decomposition)
- various kinds of spatial analysis are easy Find all streets in forests in BW
  example: overlay

theme 1 streets
theme 2 forests
theme 3 streets
result

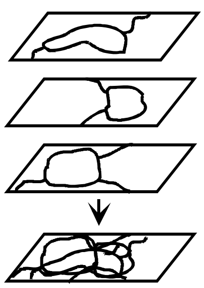


stacking of 2dimensional arrays

VECTOR

- point, lines, areas = real world object
  abstraction of the geometry of the real world object (construction)
- overlay operations are difficult to implement
  example: overlay

theme 1
theme 2
theme 3
result



calculation of intersections

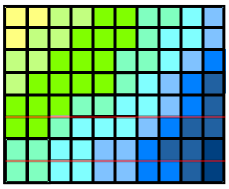
minimal common geometry

Raster Data vs. Vector Data

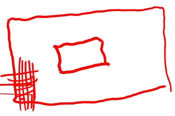
RASTER | ADVANTAGES / DISADVANTAGES | VECTOR

efficient structure for high spatial variability

examples: temperature or elevation

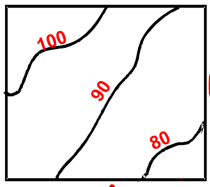


not efficient for low variability



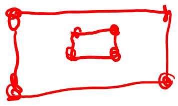
inefficient structures for high spatial variability

examples: temperature or elevation



isolines + interpolation algorithm

efficient low variability



Raster Data vs. Vector Data

selection depends on the application

RASTER

- areal applications with high spatial variability and less demands on precision

weather forecast

VECTOR

- linear application with low spatial variability and demand on high precision

cadastre

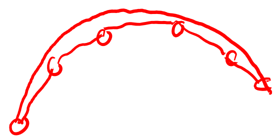
ifp

Universität Stuttgart

Vector World: Basic Elements

Point:
- ID
- coordinates x, y
- Type
- Quality measure: reliable, point errors, date of collection

Line:
- ID
- list point IDs min. 2
- Type
- Direction
- Type of connection



---

ifp

Universität Stuttgart

2D

Polygon/Area:
- ID
- list line ID
- Type

2,5D

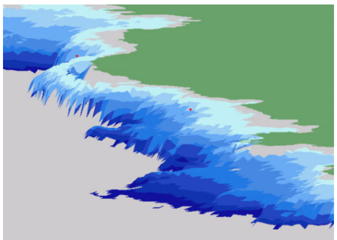
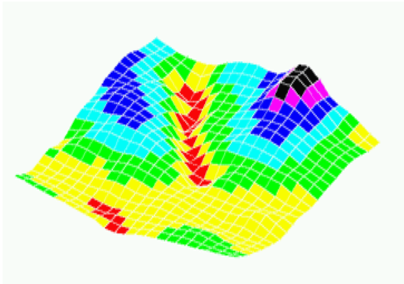
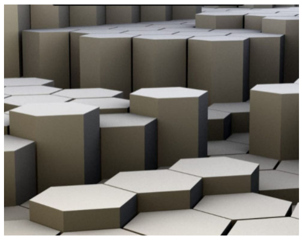
Volumes: many different modelling strategies

3D

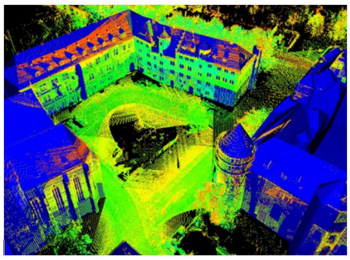
ifp

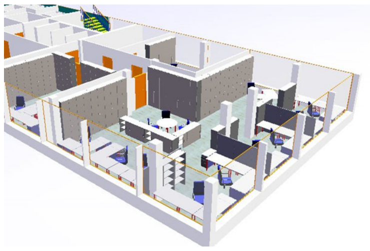
2.5D and 3D Data

- 2.5D: for every Position (*x, y*) exactly one *z*-Value (Surfaces):

- 3D: one (*x, y*) Position can have several *z*-Values (Bodies or Solids):



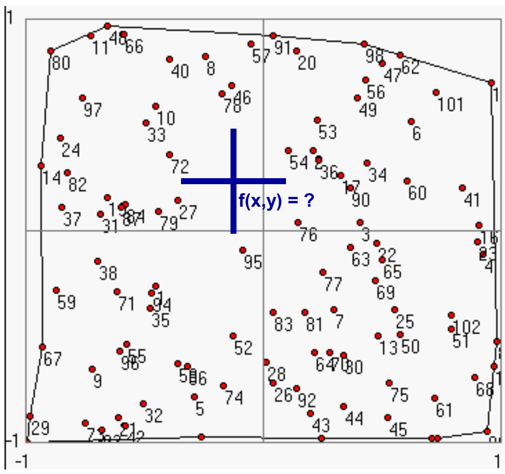


Universität Stuttgart

ifp

Modeling of Surfaces

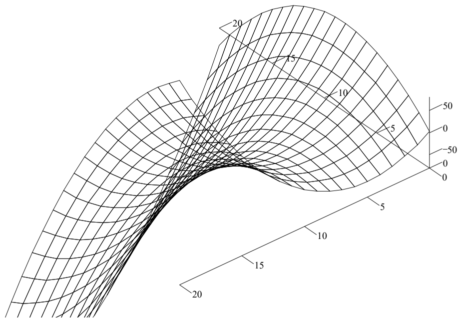
- Problem
  - Given: point cloud
    - Height data
    - Waether data
    - Pollution data
    - .....
  - Searched is the z-value at any position x, y



Universität Stuttgart

Surface Reconstruction with Polynoms

- Single-valued surface f(*x*, y): $\Re^2 \rightarrow \Re$
  - Approximation of the 2.5D points with a polynom:
    $z = f(x,y) = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2 + \ldots$



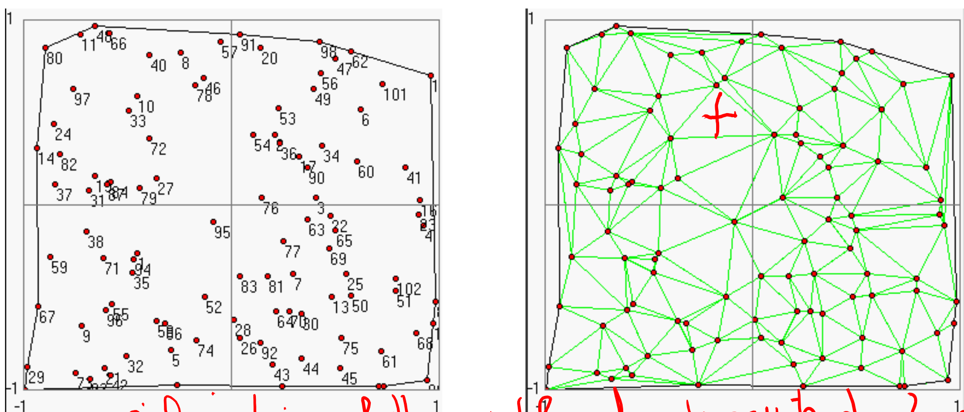
- Problem: only simple forms are possible

Universität Stuttgart

Surface Reconstruction with Irregular Triangulated Networks (TIN)

- Neighbored points are connected in order to get triangles
- The height at a position (*x*, *y*) is calculated by interpolating in the corresponding triangle

2 Steps: 1) Triangulation 2) Interpolation in Triangles

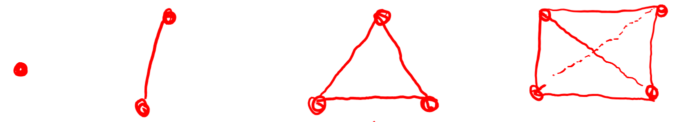


Simplification of the problem from 4 points to 3 points

Universität Stuttgart

ifp

Universität Stuttgart



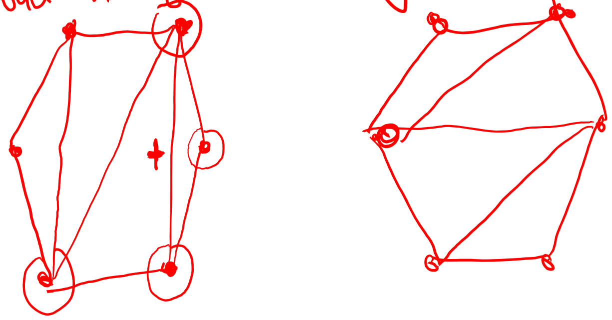
no real solution

simple solution

several solutions

„bad" triangulation

good triangulation



Criteria

→ angles near 60 degrees

ifp

Universität Stuttgart

Brute Force Triangulation

1	100	101
2	99	101
3	98	101

} 50

$\frac{n(n+1)}{2}$

$\approx n^2$

Brute Force Algorithms:

+ easy to implement
+ good results
- high computation time

Triangulation algorithm

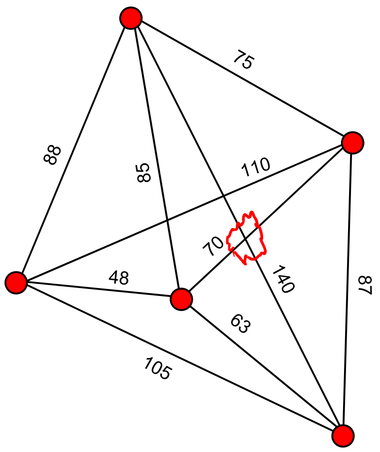
- Calculate all distances between points $\frac{n(n-1)}{2}$
- Search for minimal distance and delete all distances that intersect this distance
- Continue until no intersections exist

n	n^2
1 000	1 000 000
10 000	100 000 000
100 000	10 000 000 000

ifp

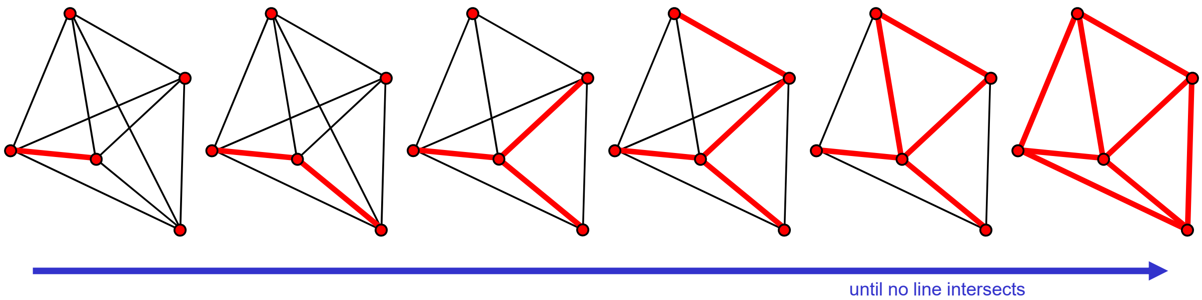
Universität Stuttgart

Brute Force Triangulation



$n = 5$

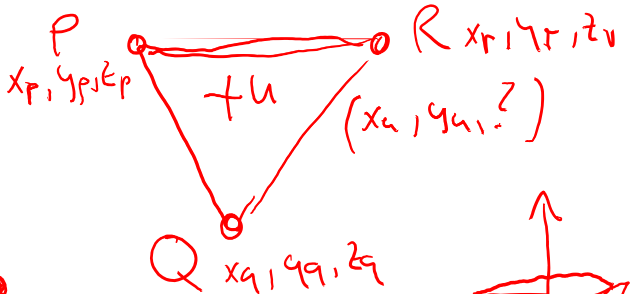
$$\frac{n(n-1)}{2} = 10$$

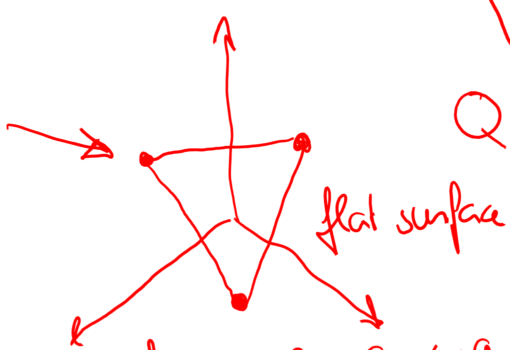


ifp

Universität Stuttgart

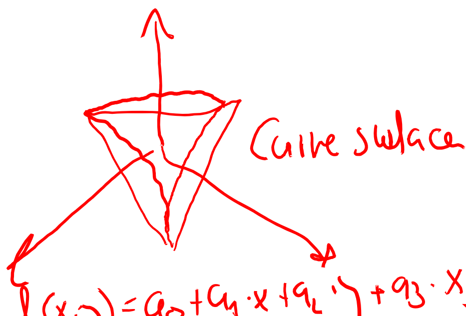
Interpolation in Triangles





$f(x,y) = a_0 + a_1 \cdot x + a_2 y$

Linear Interpolation



$f(x,y) = a_0 + a_1 \cdot x + a_2 \cdot y + a_3 \cdot xy$

Cubic Interpolation

ifp

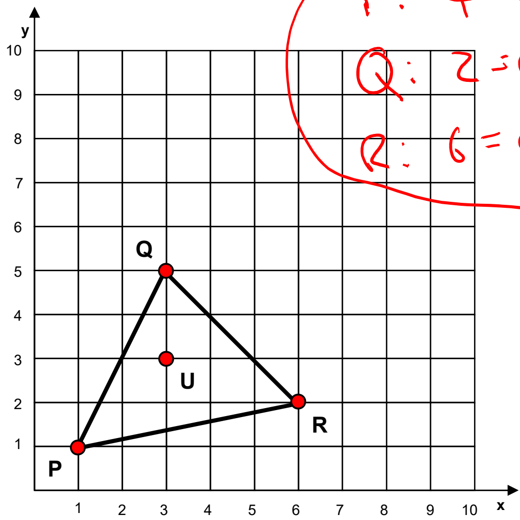
Lineare Interpolation in Triangles Example

$z = f(x,y) = a_0 + a_1 \cdot x + a_2 \cdot y$

$P: \; 4 = a_0 + a_1 \cdot 1 + a_2 \cdot 1$

$Q: \; 2 = a_0 + a_1 \cdot 3 + a_2 \cdot 5$

$R: \; 6 = a_0 + a_1 \cdot 6 + a_2 \cdot 2$



P = (1, 1, 4)
Q = (3, 5, 2)
R = (6, 2, 6)
U = (3, 3, ?)

Universität Stuttgart

ifp

z = f(x, y) = a0 + a1x + a2y single valued surface
P (1, 1, 4): 4 = a0 + 1 a1 + 1 a2 (1)
Q (3, 5, 2): 2 = a0 + 3 a1 + 5 a2 (2)
R (6, 2, 6): 6 = a0 + 6 a1 + 2 a2 (3)

(1): a0 = 4 – a1 – a2
(2): a0 = 2 – 3a1 – 5a2

(1) = (2):
4 – a1 – a2 = 2 – 3a1 – 5a2
2a1 = – 2 – 4a2
a1 = –1 – 2a2 (4)

(2): a0 = 2 – 3a1 – 5a2
(3): a0 = 6 – 6a1 – 2a2

(2) = (3):
2 – 3a1 – 5a2 = 6 – 6a1 – 2a2
3a1 = 4 + 3a2 (5)

3*(4) = (5):
-3 – 6a2 = 4 + 3a2
9a2 = -7
a2 = -7/9 (6)

(6) in (4):
a1 = -1 –2(-7/9)
a1 = 5/9 (7)

(6),(7) in (1):
a0 = 4 – 5/9 + 7/9
a0 = 38/9

U (3, 3, ?):
U = a0 + 3a1 + 3a2
U = 38/9 + 15/9 – 21/9
U = 32/9

Universität Stuttgart

ifp

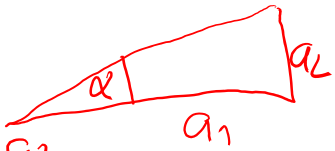
Universität Stuttgart

$$z = f(x,y) = a_0 + a_1 \cdot x + a_2 \cdot y$$

gradient in x-direction $\partial z / \partial x = a_1$

" y - " $\partial z / \partial y = a_2$

total gradient $\sqrt{a_1^2 + a_2^2}$



total gradient direction $\tan \alpha = \frac{a_2}{a_1}$

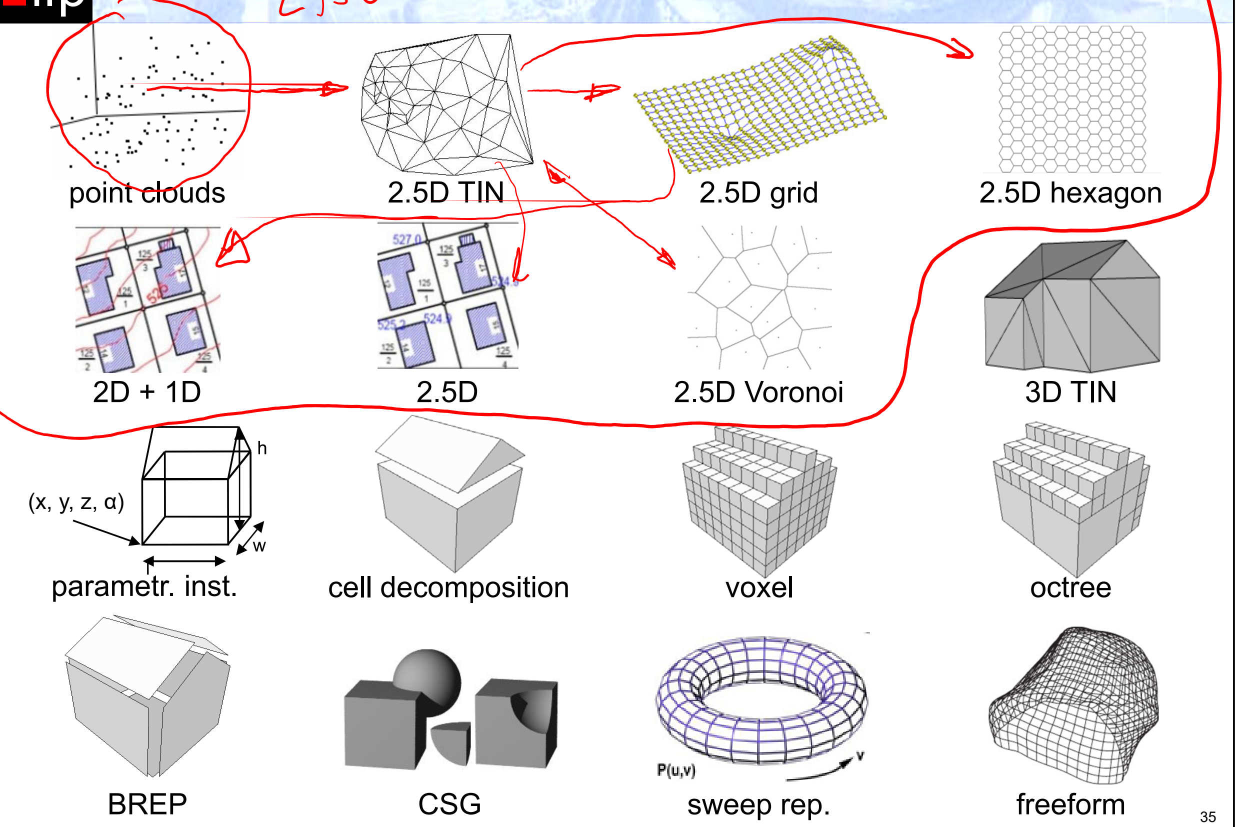
ifp

Universität Stuttgart

TIN

Data Modelling in 3D is much more complex ...

2,5D



Universität Stuttgart

Geometric Modelling of 3D Objects

- Basic 3D Models:
  - Primitive Instancing
  - Cell Decomposition
  - Spatial Occupancy Enumeration
  - Boundary Representation (BREP)
  - Constructive Solid Geometry (CSG)
  - Sweep Representations
  - Freeform Shapes
  - …

Universität Stuttgart

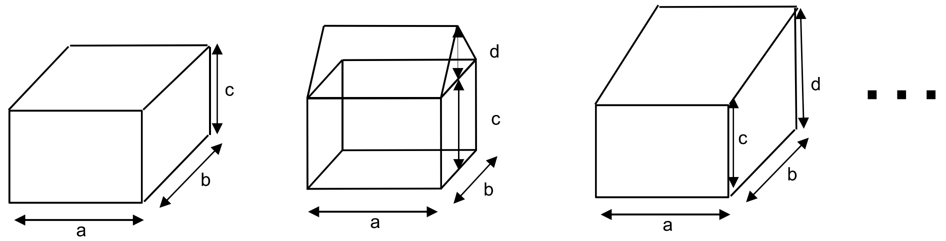
Primitive Instancing

ifp

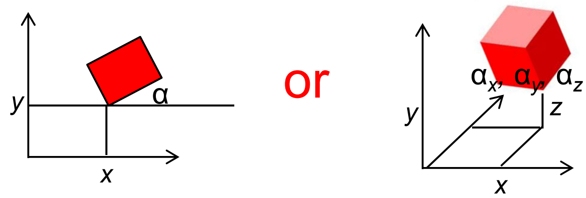
Universität Stuttgart

- Define for each Object Type a complete List of Description Parameters
- We need Parameters to describe the **Form** of an Object:

object type: flat roof | **object type: sloped roof house** | **object type: pitch roof house**



and the **Position**:



Primitive Instancing

ifp

Universität Stuttgart

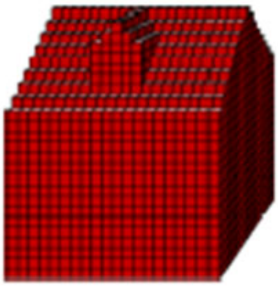
- **Pro**: Simple model, low Data Volume
- **Contra**: You can use only predefined Models -> it is not possible to combine Instances to create new more complex Structures
- In GIS it was used in the Past (15 – 20 years ago) for the Modelling of Houses because at this Time it was only possible to reconstruct simple Objects (Houses) from Point Clouds
  - 20 Models can describe more than 90% of all Houses (in rural Areas)
  - Complex Houses are difficult to describe (inner-city Areas)
- It is the basic Technique for Cell Decomposition and Constructive Solid Modelling (see later)

ifp

Spatial Occupancy Enumeration

Universität Stuttgart

- A 3D Object is modeled by a List of spatial Cells occupied by the Object
- The Cells (or Voxels) are Cubes of a fixed Size and are arranged in a fixed Spatial Grid.
- This Method is analog to raster Modelling in 2D
- **Pro**: Simple Structure (a list of occupied Cells). It is easy to determine if a Point lies inside or outside of an Object.
- **Contra**: Higher Model Accuracy requires higher Number of Cells -> huge Data Volumes

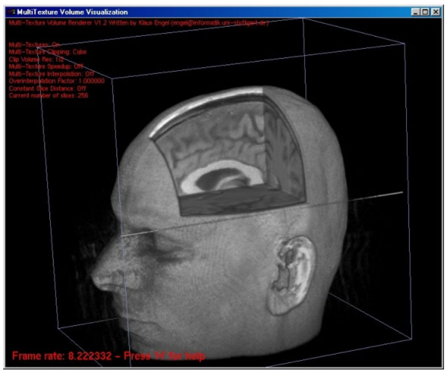


ifp

Spatial Occupancy Enumeration

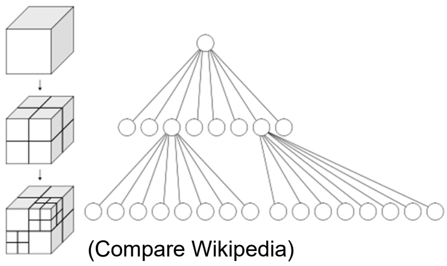
Universität Stuttgart

- Representation used primarily for Volume Visualization
- Popular for medical Purposes such as
  - CAT scans
  - Magnetic Resonance Imaging (MRI)

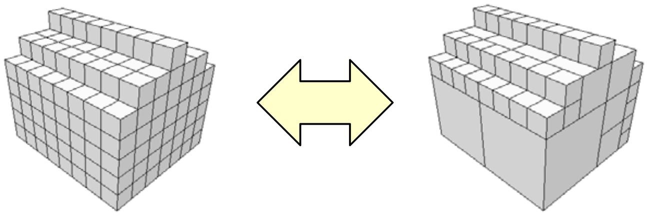


Octtree

- An Octree is a Tree Data Structure in which each (internal) Node has eight Children.
- The Leafs are analog to Voxels in Spatial Occupancy Enumeration
- Octrees are the three-dimensional analog to Quadtrees.
- **Pro**: less Data Volume as Spatial Occupancy Enumeration
- **Contra**: access to the Data is not so easy as in Spatial Occupancy Enumeration



(Compare Wikipedia)



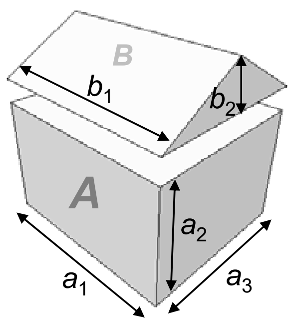
Spatial Occupancy Enumeration

Octree

Cell Decomposition

- *Cell Decomposi*tion is an Extension of *Spatial Occupancy Enumeration*
- The Cells are not only Cubes but also other Primitives like Prisms, Spheres, Cylinders, Cones, …
- The Cells can optionally be parameterized with *Primitive Instancing*

Cell Decomposition without *Primitive Instancing* = LEGO

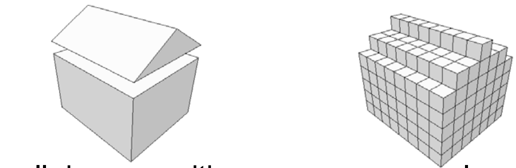


Cell Decomposition with *Primitive Instancing*

ifp

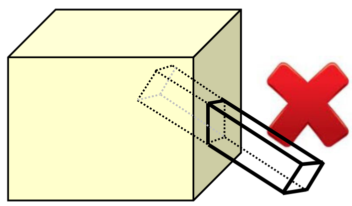
Cell Decomposition

- **Pro**: better Representation of 3D Objects (no blocky Structures because of Voxels)



cell decomposition   voxel

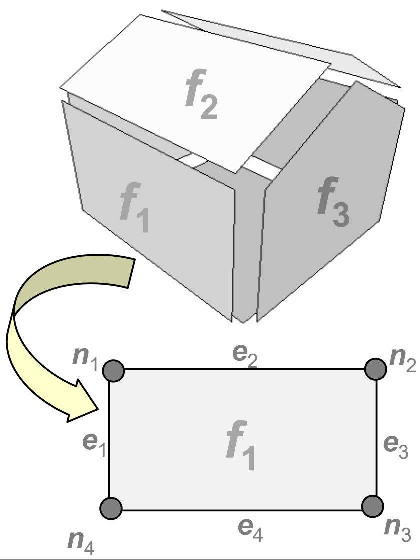
- **Contra**: more difficult to guarantee Correctness (Cells must fit together – it is not allowed that one Cell penetrates other Cells)



ifp

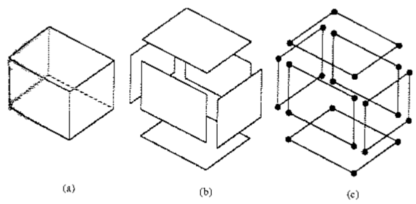
Boundary Representation (BREP)

- 3D Objects are defined by their enclosing Surfaces
- Topological Representation: an Object consist of *n* Faces – a Face consist of *n* Edges – an Edge consist of two Nodes – a Node has one Coordinate



Boundary Representation (BREP)

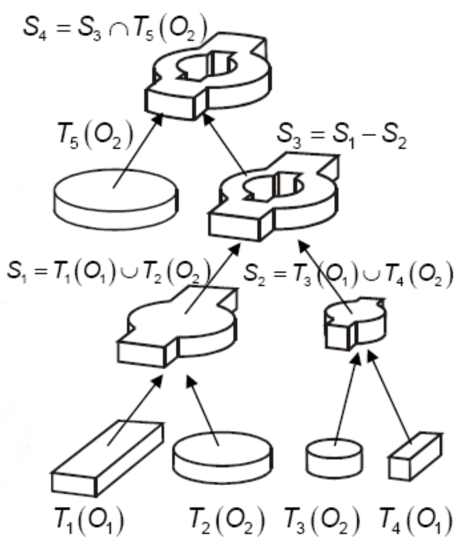
- Commonly used in GIS (e.g. CityGML use BREP)
- Also often used in CAD Systems
- **Pro**:
  - very flexible
  - direct Extension of 2D Vector Data
  - it is possible to make local Changes without complete new Construction
  - Topology is completely stored -> Topological Analyses are possible
- **Contra**:
  - Correctness is difficult to prove
  - High Data Volumes



Constructive Solid Geometry (CSG)

- CSG uses Primitives (Prisms, Spheres, Cylinders, Cones, etc.) and Boolean Operations (Union, Subtraction, Intersection) to create 3D Objects
- The Primitives are modeled with Primitive Instancing

CSG-Tree:



Constructive Solid Geometry (CSG)

- Often used in CAD systems but also in GIS or in Game Engines
- **Pro:**
  - Easy to construct very complex Models with few Primitives
  - CSG Modelling need less Storage.
  - CSG can be converted to BREP
- **Contra:**
  - It is not unique (one 3D Object can have many different CSG Trees)
  - The Visualization of CSG Objects is CPU intensive

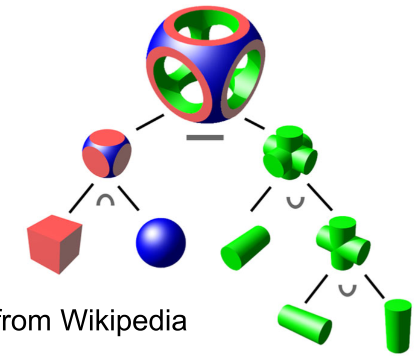
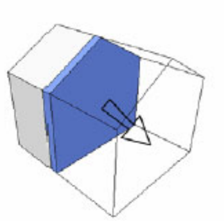


Image from Wikipedia

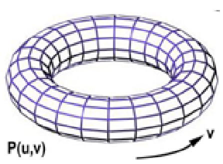
Universität Stuttgart

Sweep Representations

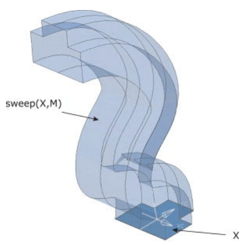
- Sweep Representations are used to construct 3D Objects that have some kind of Symmetry
- A Sweep Representation consists of a Shape and a Trajectory



Translational Sweep



Rotational Sweep



Complex Sweep

- Variations:
  - Vary the Shape along the Sweep Path
  - Vary the Orientation of the Shape relative to the Sweep Path.
- Sweep Representations allow the Modelling of very Complex Objects
- Used in many CAD Systems

Universität Stuttgart

Freeform Surfaces

- Freeform surfaces are used to describe the Skin of 3D objects
- Most Systems today use nonuniform rational B-Splines (NURBS) Mathematics to describe the Surface Forms
- Used in CAD Systems