

Triangulieren eines Polygons

Kann man jedes (einfache) Polygon triangulieren?

Meisters „Zwei-Ohren-Theorem“:

Jedes einfache Polygon P mit mindestens 4 Vertizes hat mindestens zwei „Ohren“, d.h. Dreiecke aus drei benachbarten Vertizes, die vollständig in P liegen

Meisters, G.H. Polygons have ears. *American Mathematical Monthly*. June/July 1975, 648-651.

Triangulieren eines Polygons

Kann man jedes Polygon triangulieren?

Trivialer Algorithmus: „Ohren abschneiden“:

Konvexen Vertex v_i finden, Nachbarn v_{i+1} , v_{i-1} bestimmen,
Sicherstellen, dass kein weiterer Vertex v im Dreieck (v_i, v_{i+1}, v_{i-1}) liegt

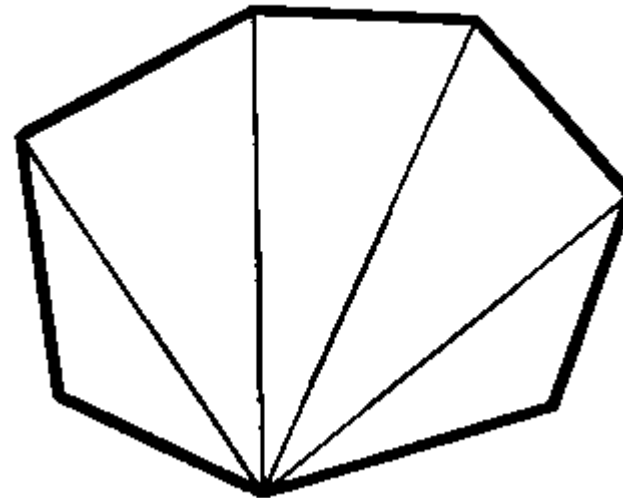
Komplexität?

Geht's besser?

Triangulieren eines Polygons

Kann man jedes Polygon triangulieren?

Konvexe Polygone: trivial



Beliebige einfache Polygone?

Triangulieren eines Polygons

Kann man jedes Polygon triangulieren?

[DeBerg, Overmars: Computational Geometry]

Hier nur grobe Beweisskizze für den Nachweis der Komplexität

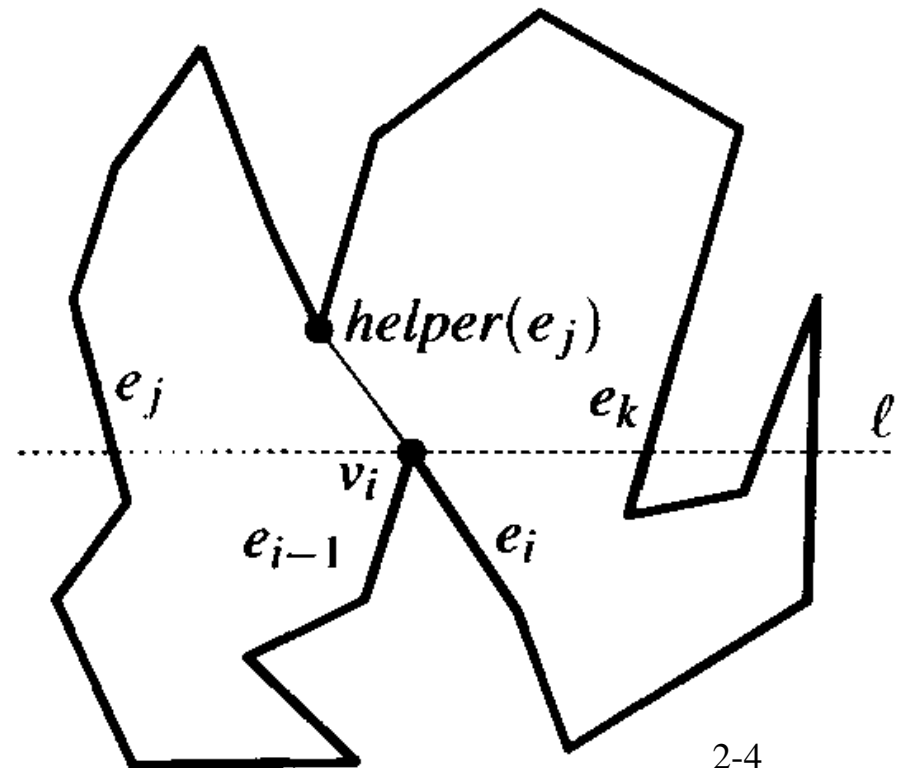
Beliebige Polygone:

In monotone Polygone aufteilen,
(nicht ganz trivial)

dann diese triangulieren

Komplexität der Zerlegung:

$O(n \log(n))$



Triangulieren eines Polygons

Kann man jedes Polygon triangulieren?

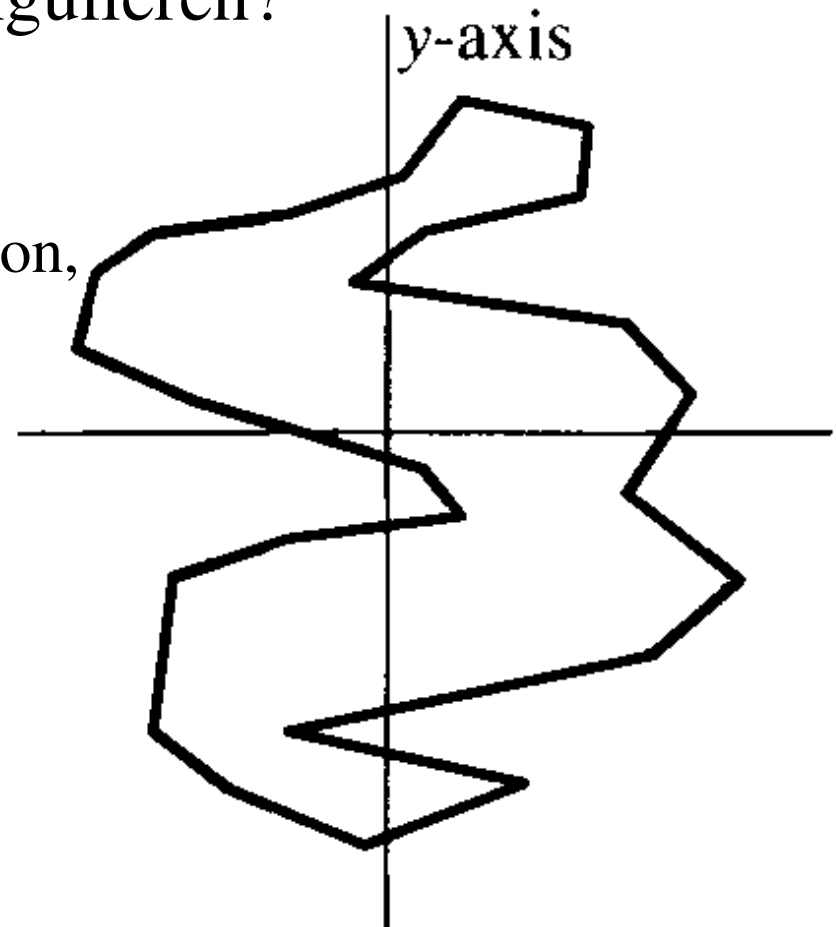
[DeBerg, Overmars: Computational Geometry]

Monotone Polygone:

Vertizes monoton in einer Dimension,

Jede Linie senkrecht zu dieser
schneidet das Polygon

0, 1, 2 mal oder in einer Kante
(keine „unzusammenhängende“
Aufteilung)



Triangulieren eines Polygons

Kann man jedes Polygon triangulieren?

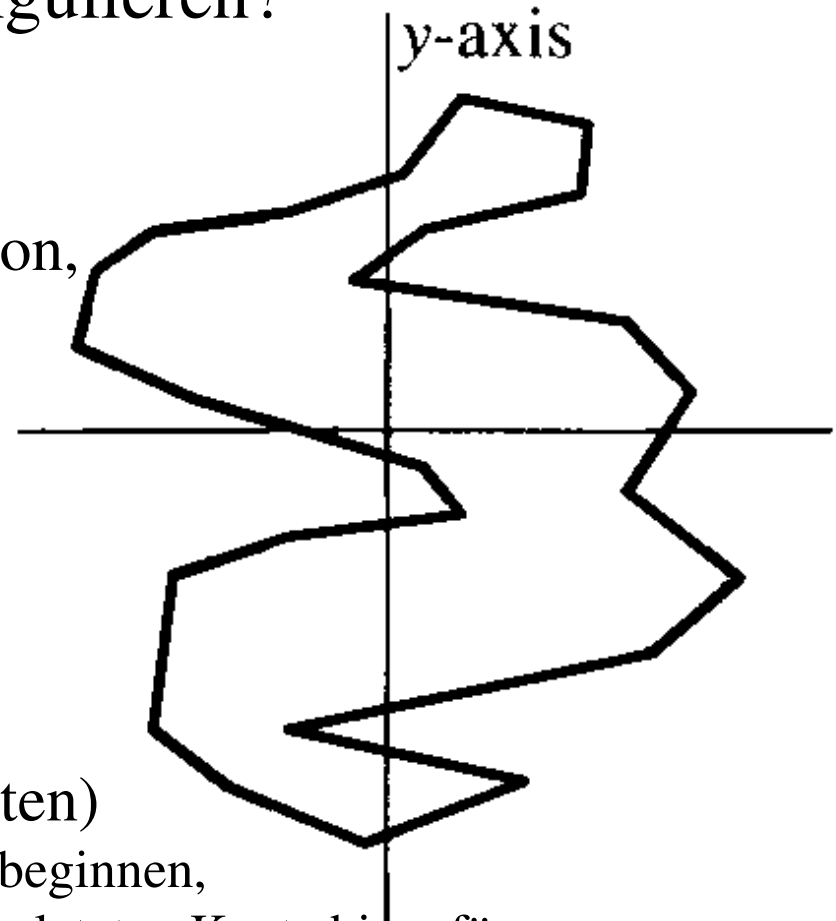
[DeBerg, Overmars: Computational Geometry]

Monotone Polygone:

Vertizes monoton in einer Dimension,

Jede Linie senkrecht zu dieser
schneidet das Polygon

0, 1, 2 mal oder in einer Kante
(keine „unzusammenhängende“
Aufteilung)



Triangulierung: (Von oben oder unten)

- mit Dreieck aus den ersten drei Punkten beginnen,
- jeweils nächsten Punkt der Sortierung zur letzten Kante hinzufügen

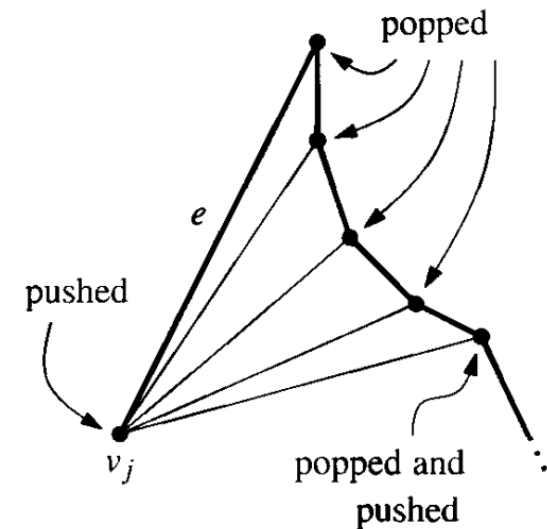
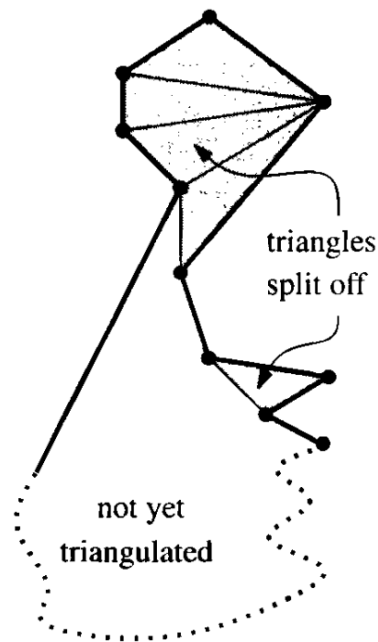
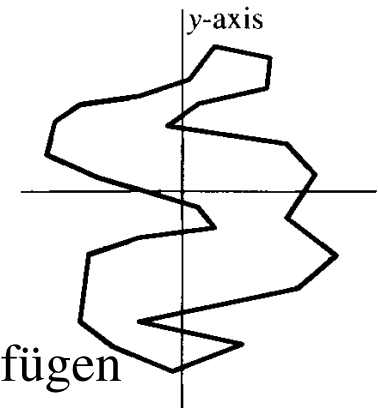
Triangulieren eines Polygons

Kann man jedes Polygon triangulieren?

[DeBerg, Overmars: Computational Geometry]

Triangulierung: (Von oben oder unten)

- mit Dreieck aus den ersten drei Punkten beginnen,
- jeweils nächsten Punkt der Sortierung zur letzten Kante hinzufügen
- Stack für nicht verarbeitbare Punkte
- Komplexität $O(n)$!



Triangulieren eines Polygons

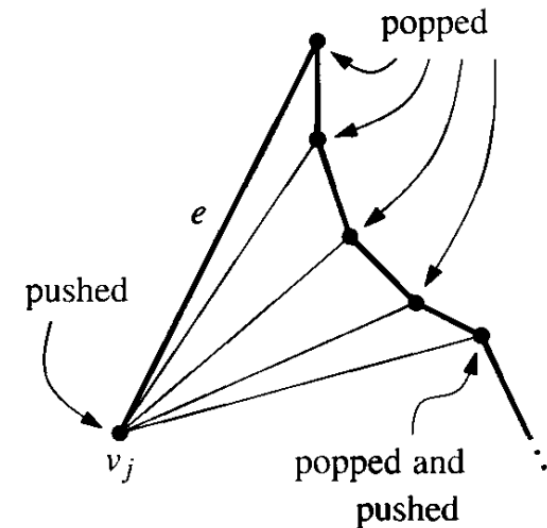
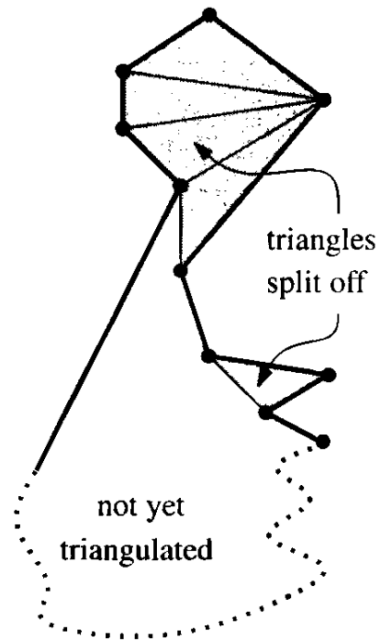
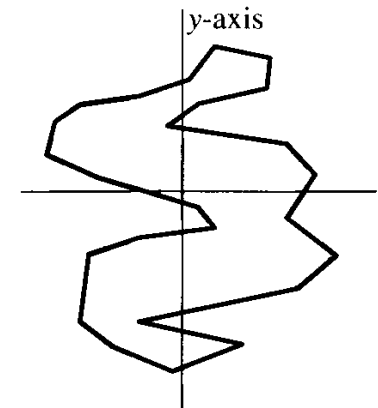
Kann man jedes Polygon triangulieren?

[DeBerg, Overmars: Computational Geometry]

Zerlegen in monotone Polygone: $O(n \log(n))$

Monotones Polygon triangulieren: $O(n)$

Summe: $O(n \log(n))$



Triangulieren eines Polygons

Pseudocode (der Vollständigkeit halber)

[DeBerg, Overmars: Computational Geometry]

Algorithm TRIANGULATEMONOTONEPOLYGON(\mathcal{P})

Input. A strictly y -monotone polygon \mathcal{P} stored in a doubly-connected edge list \mathcal{D} .

Output. A triangulation of \mathcal{P} stored in the doubly-connected edge list \mathcal{D} .

1. Merge the vertices on the left chain and the vertices on the right chain of \mathcal{P} into one sequence, sorted on decreasing y -coordinate. If two vertices have the same y -coordinate, then the leftmost one comes first. Let u_1, \dots, u_n denote the sorted sequence.
2. Initialize an empty stack S , and push u_1 and u_2 onto it.
3. **for** $j \leftarrow 3$ **to** $n - 1$
4. **do if** u_j and the vertex on top of S are on different chains
5. **then** Pop all vertices from S .
6. Insert into \mathcal{D} a diagonal from u_j to each popped vertex, except the last one.
7. Push u_{j-1} and u_j onto S .
8. **else** Pop one vertex from S .
9. Pop the other vertices from S as long as the diagonals from u_j to them are inside \mathcal{P} . Insert these diagonals into \mathcal{D} . Push the last vertex that has been popped back onto S .
10. Push u_j onto S .
11. Add diagonals from u_n to all stack vertices except the first and the last one.

Flächeninhalte und Volumen

Flächeninhalte und Volumen

Flächeninhalt eines Dreiecks???

$$F = \frac{1}{2} |a| \cdot |h| = \frac{1}{2} |a| \cdot |b| \cdot \sin(\theta)$$

Kreuzprodukt !!! (2D: Einbettung
in 3D)

$$F = \frac{1}{2} |a \times b|$$

oder mit ccw und Betrag

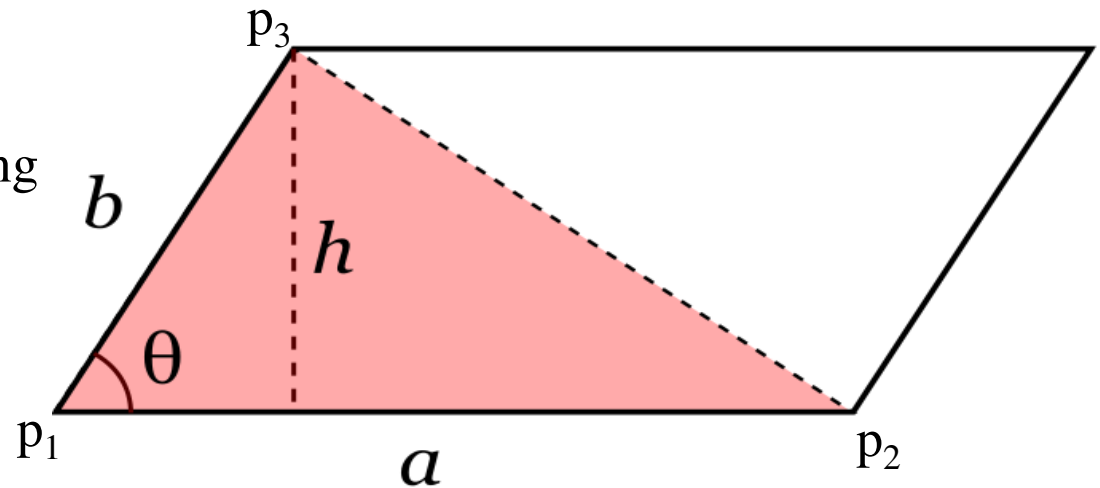
$$F = \frac{1}{2} |\text{ccw}(p_1, p_2, p_3)|$$

oder mit Rücksicht auf cw, ccw

$$F = \frac{1}{2} \text{ccw}(p_1, p_2, p_3) \text{ (positiv, wenn ccw; negativ, wenn cw)}$$

oder mit $p_1 = 0$:

$$F = \frac{1}{2} \text{ccw}(0, a, b) = \frac{1}{2} (a_x b_y - a_y b_x)$$



Flächeninhalte und Volumen

Flächeninhalt eines Polygons???

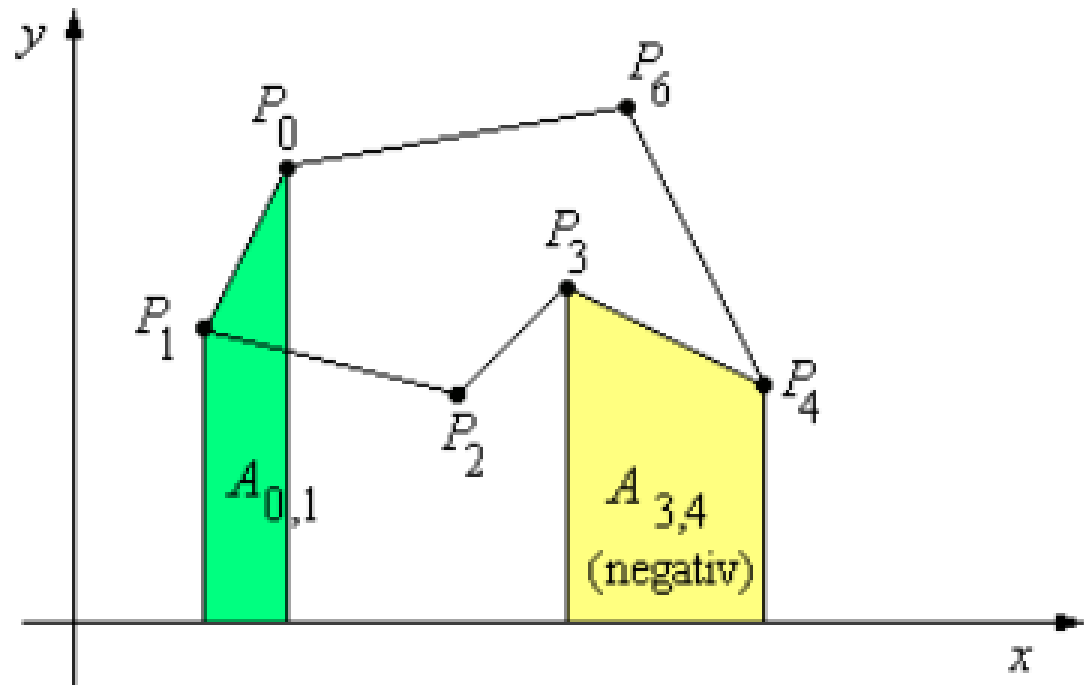
Flächeninhalte und Volumen

Flächeninhalt eines Polygons???

Gaußsche Trapezformel

- Trapeze mit positiven und negativen Flächeninhalten

$$A = \sum_{i=1}^n (y_i + y_{i+1})/2 \cdot (x_i - x_{i+1})$$



Flächeninhalte und Volumen

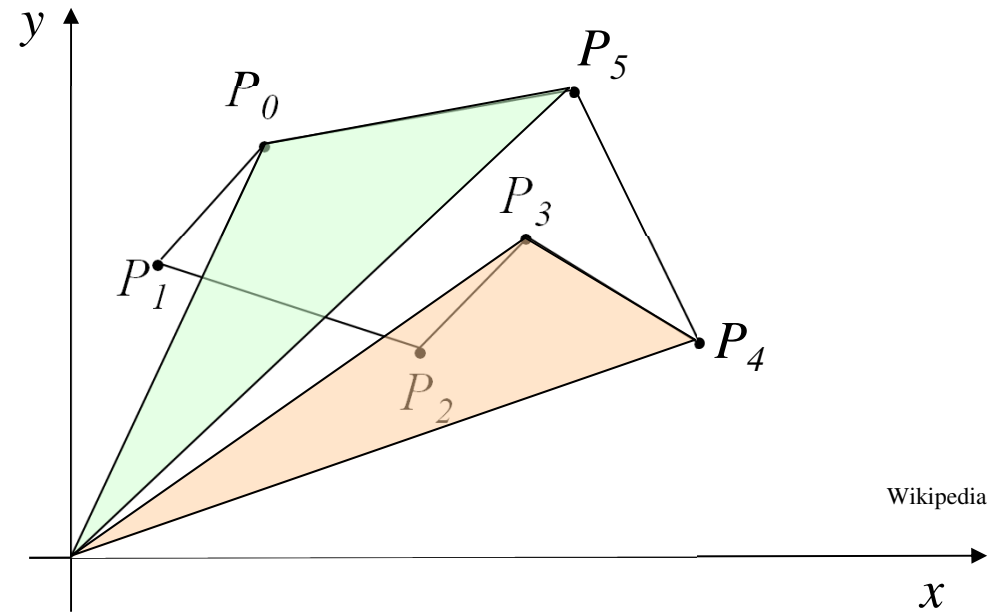
Flächeninhalt eines Polygons???

Flächeninhalte und Volumen

Flächeninhalt eines Polygons???

Gaußsche Dreiecksformel: Analog mit Dreiecken $(0, P_n, P_{n+1})$

$$\begin{aligned} A &= \sum_{i=1}^n (y_{i+1}x_i - y_i x_{i+1})/2 \\ &= \sum_{i=1}^n (y_i x_{i-1} - y_i x_{i+1})/2 \\ &= \sum_{i=1}^n y_i (x_{i-1} - x_{i+1})/2 \end{aligned}$$



Flächeninhalte und Volumen

Volumen in 3D???

Analogie: Dreieck \rightarrow Tetraeder, Polygon \rightarrow Polyeder

Volumen eines Tetraeders?

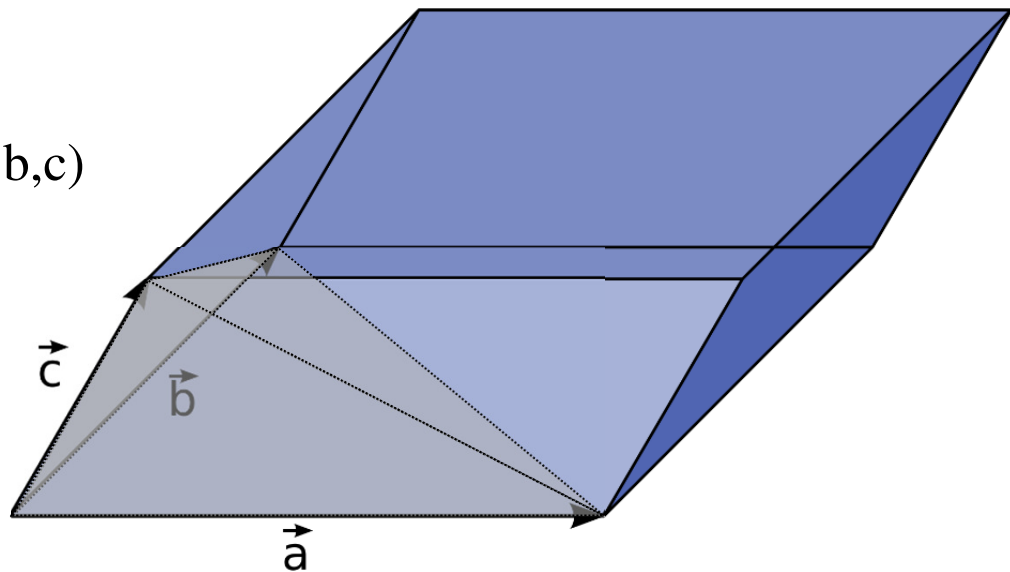
Erstmal Volumen Spat

Spatprodukt: $(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}$ bzw. $\det(\mathbf{a}, \mathbf{b}, \mathbf{c})$

$$V = \det \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}$$

positiv, falls $\mathbf{a}, \mathbf{b}, \mathbf{c}$ ccw

negativ sonst.



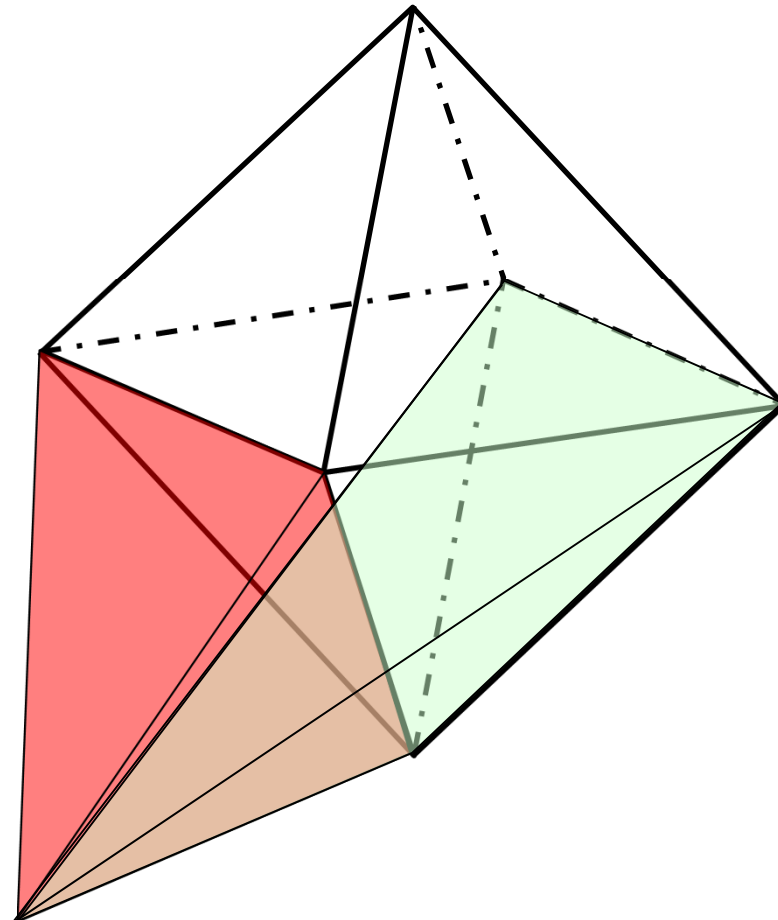
$$\text{Tetraeder: } V = 1/6 \cdot \det \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}$$

Flächeninhalte und Volumen

Volumen in 3D???

Volumen eines triangulierten Polyeders analog zu Polygonen!!!

- Aufsummieren der positiven und negativen Beiträge pro Dreieck (bzw. jeweils erzeugtem Tetraeder)



Modellierung, 2D und 3D

Kurzer Überblick über die Methoden, wie Objekt-Modelle in 2D und 3D dargestellt werden

Referenz:

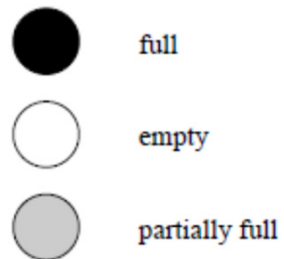
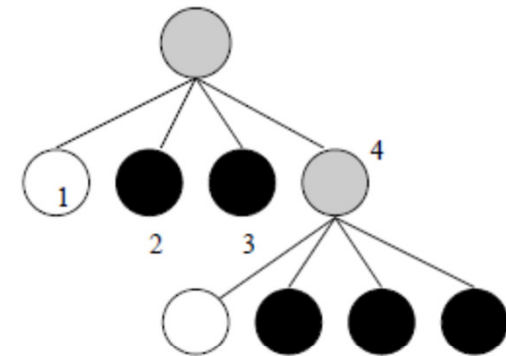
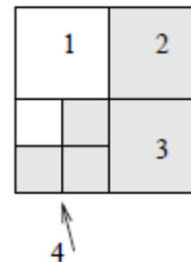
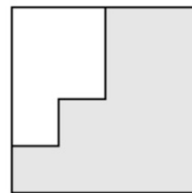
<http://ocw.mit.edu/courses/mechanical-engineering/2-158j-computational-geometry-spring-2003/lecture-notes/>

Lehrveranstaltungen 1, 14, 15

Modellierung, 2D

Zwei verbreitete Methoden

- Vektordarstellung
- Rasterdarstellung
 - Vollständig (Pixelbild)
 - Quad-Trees
(analog Octrees, dazu später)



Modellierung, 2D

Zwei verbreitete Methoden

- Vektordarstellung
- Rasterdarstellung

Umwandlung Vektor \rightarrow Raster:

Rasterung (z.B. in Grafikkarten, zur Darstellung,...)

Umwandlung Raster \rightarrow Vektor:

Vektorisierung

- häufig manuell,
- Verfolgen von Kanten, bis Linearisierungsfehler zu groß, dann neue Kante

Modellierung, 3D

Überblick

- Dekompositions-Modelle (*decomposition models*)
 - Raumunterteilung
- Konstruktive Modelle (*constructive models*)
 - Darstellung von Körpern durch Boolesche Operationen auf Primitiven
- Oberflächen-Modelle (*boundary models*)
 - Polyedermodelle aus Flächen (*faces*), Kanten (*edges*) und Ecken (*vertices*)

Modellierung, 3D

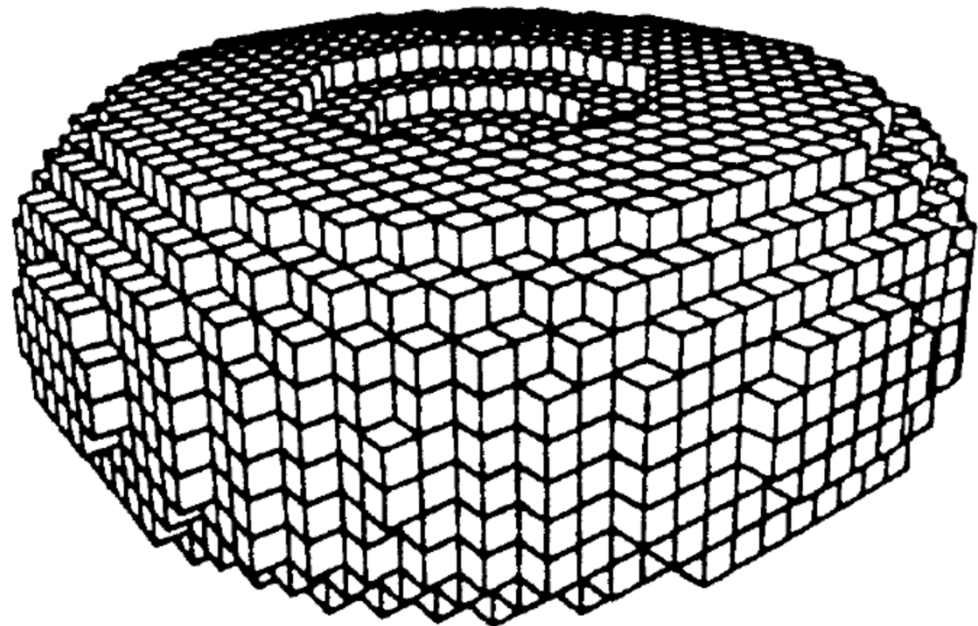
Raumunterteilung: Vollständige Aufzählung (*exhaustive enumeration*)

- Regelmäßige Raum-Unterteilung
- Volumenelemente (voxel): nicht überlappende Einheitswürfel, 3D-Rasterung

Datenstruktur: Boolesche Werte (Vom Objekt belegt, nicht belegt) in einem 3D-Array

Verwendung: Weltmodelle, medizinische Daten

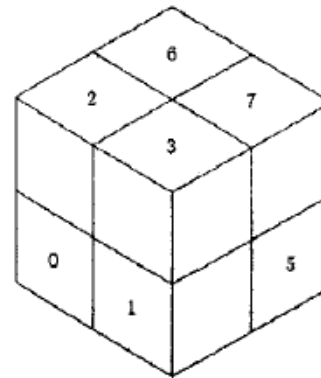
- Schnelle Zugriffe,
- Modifikation aufwändig
- Hoher Speicherbedarf



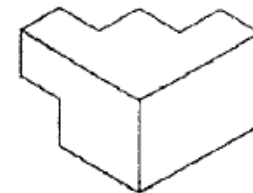
Modellierung, 3D

Raumunterteilung: Octrees

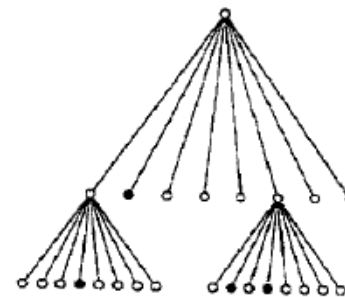
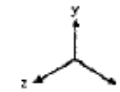
- Geringerer Speicherbedarf als 3D-Rasterung
- Approximatives Modell
- Rotationen sehr problematisch



(a)



(b)

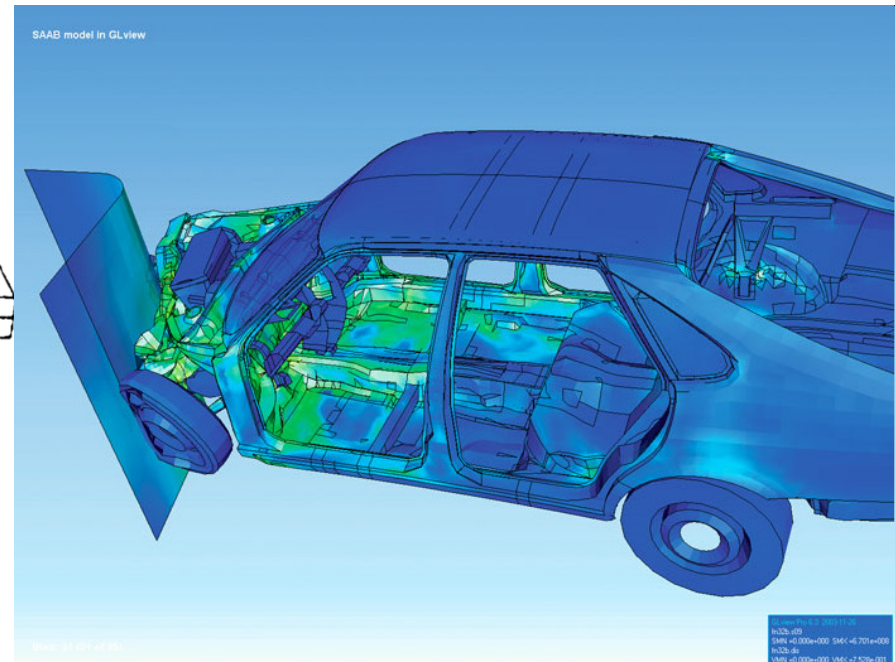
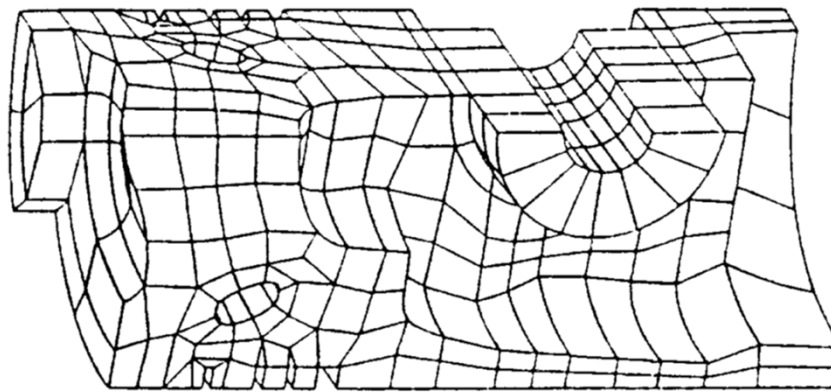


(c)

Modellierung, 3D

Raumunterteilung: Zellzerlegung

- Problemangepasste Voxel, anders als Würfel
- Verwendung: Meist Finite Element Methoden, Physikalische Simulation



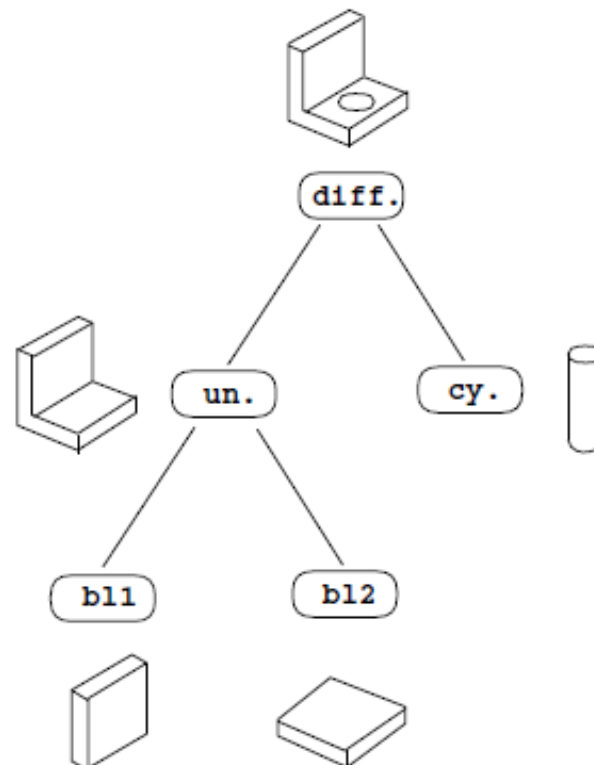
Modellierung, 3D

Konstruktive Modelle: *Constructive solid geometry* (CSG)

Basis der internen Darstellung von Objekten bei vielen CAD-Systemen

Operationen auf Volumen-Primitiven (Quader, Zylinder, Kugel, Kegel,...):

- Boolesche Operationen (Durchschnitt, Vereinigung, ...)
- Skalierung
- Translation
- Rotation



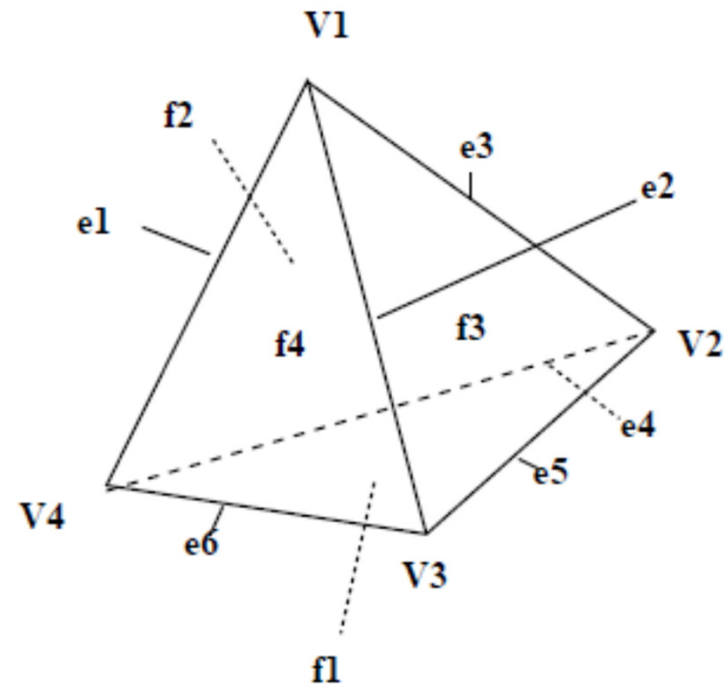
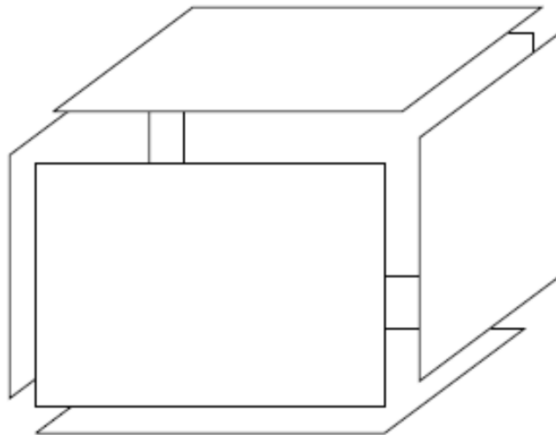
Modellierung, 3D

Oberflächenmodelle: *Boundary representation (B-Rep)*

Objekte werden durch ihre Oberflächenelemente

- Facetten, Flächen (*faces*)
- Kanten (*edges*)
- Ecken (*vertices*)

dargestellt

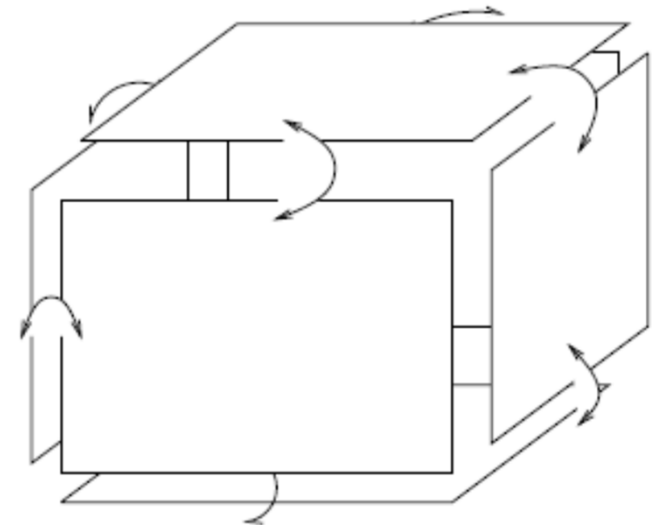


Modellierung, 3D

Oberflächenmodelle: *Boundary representation (B-Rep)*

Erforderlich:

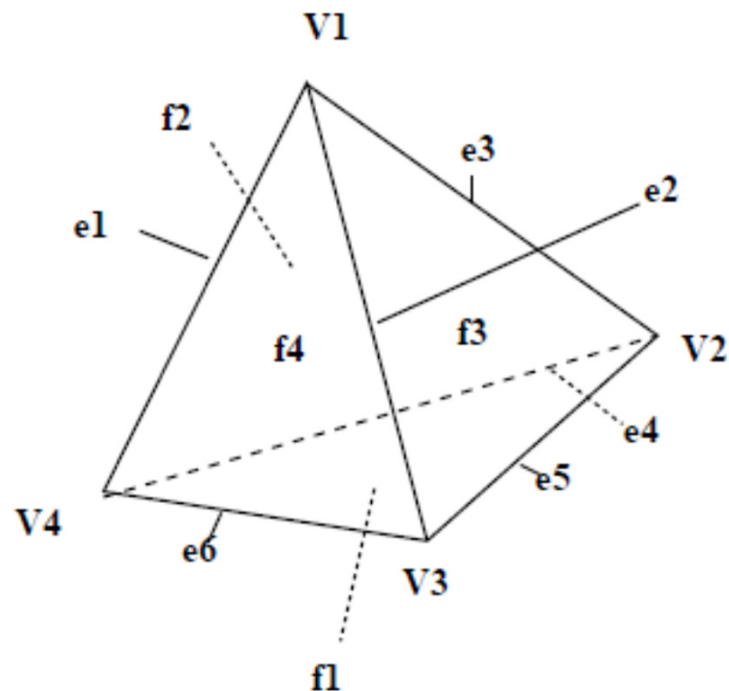
- Geometrieinformation:
 - Position der Ecken (*vertices*)
- Topologieinformation
 - Aus welchen Ecken besteht eine Kante
 - Aus welchen Kanten besteht eine Facette
 - Aus welchen Facetten besteht ein Objekt (Shell)
 - Welche Facetten stoßen an welcher Kante zusammen
 - Welche Facetten stoßen an welcher Ecke zusammen



Modellierung, 3D

Winged Edge Datenstruktur

Beispiel Tetraeder



VL Computational Geometry
Prof. M. Fischer, HS München

Object definition (region)

Boundary definition

Faces:

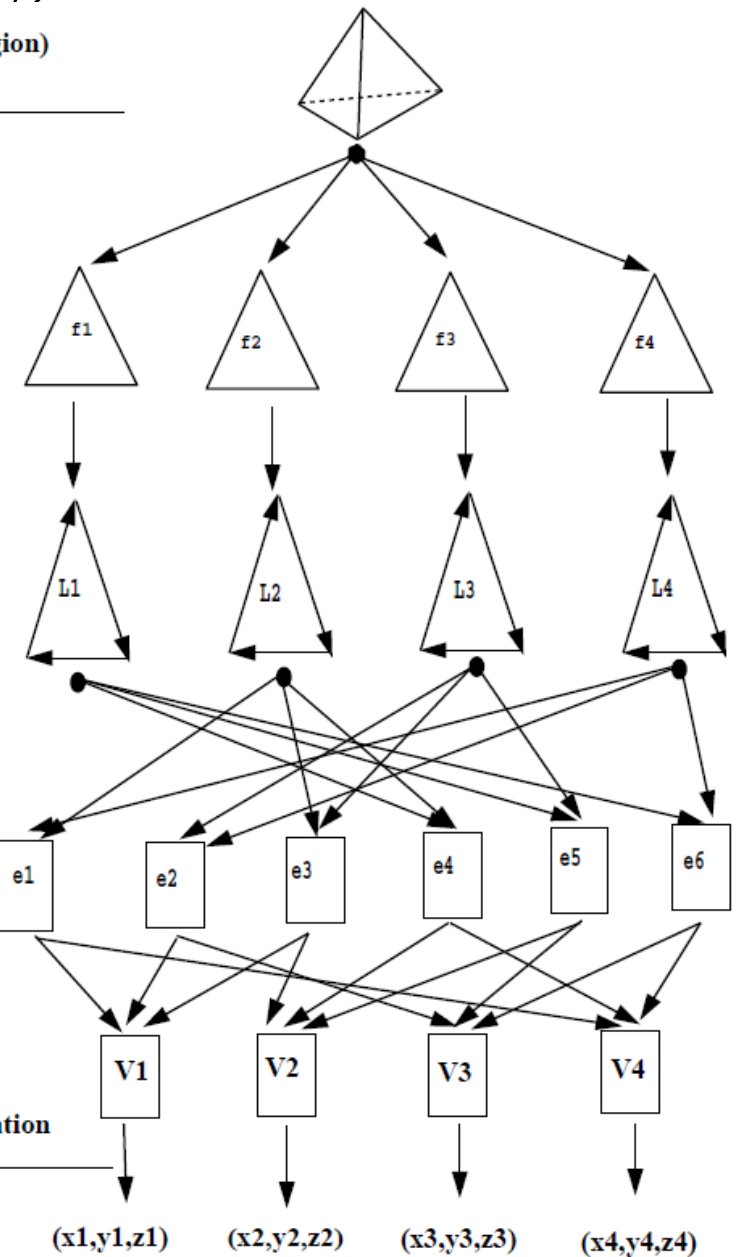
Loops:

Edges:

Vertices:

Topological Information

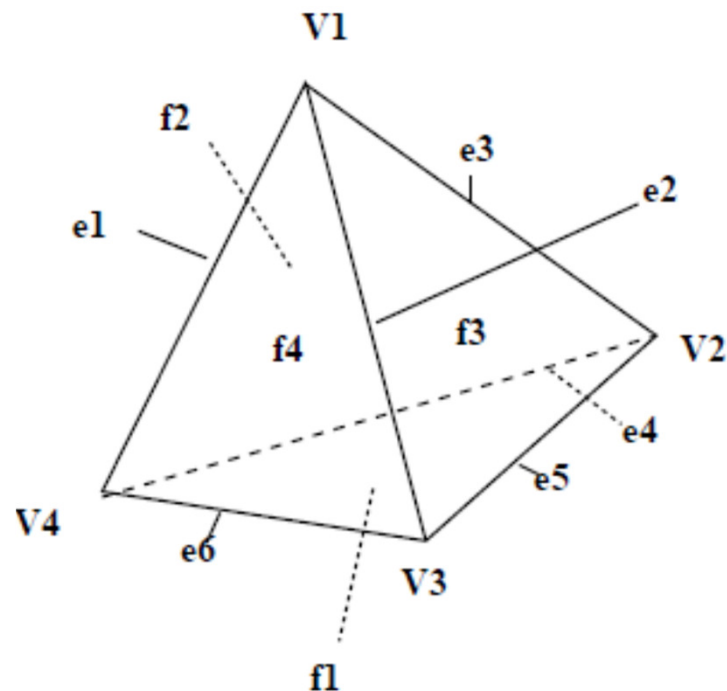
Vertex assignment
(Geometry)



Modellierung, 3D

Winged Edge Datenstruktur

Beispiel Tetraeder

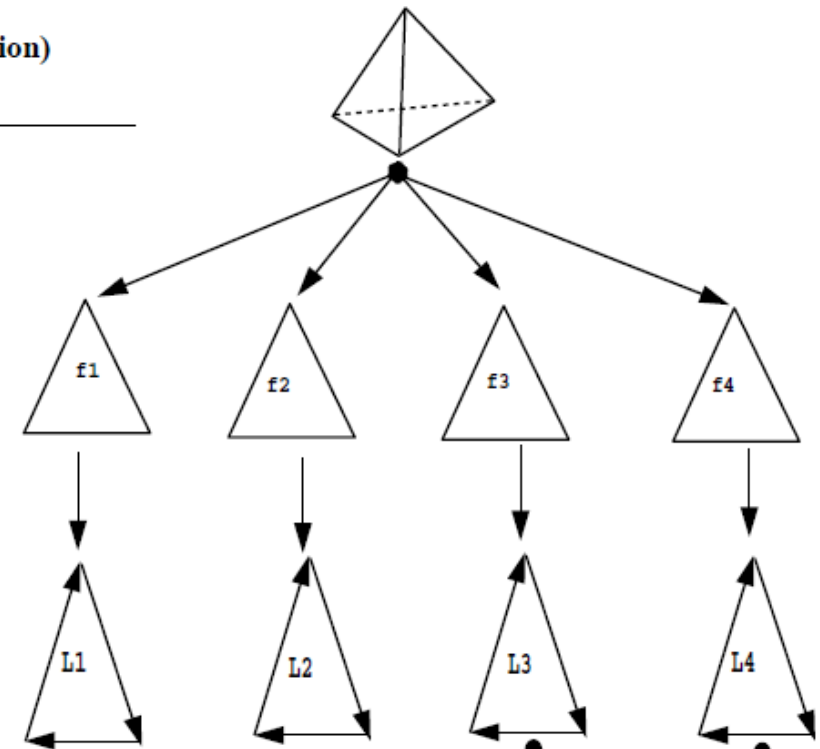


Object definition (region)

Boundary definition

Faces:

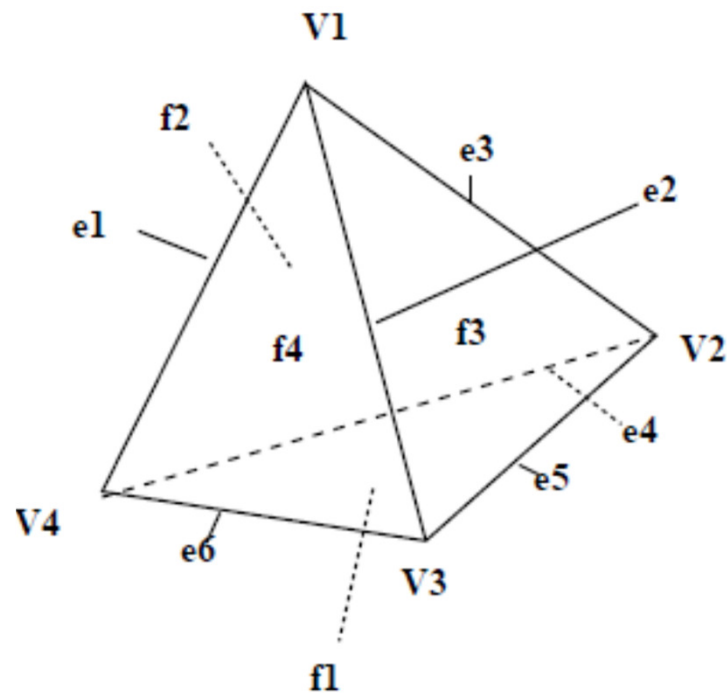
Loops:



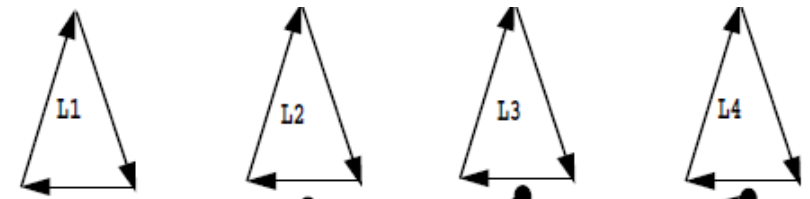
Modellierung, 3D

Winged Edge Datenstruktur

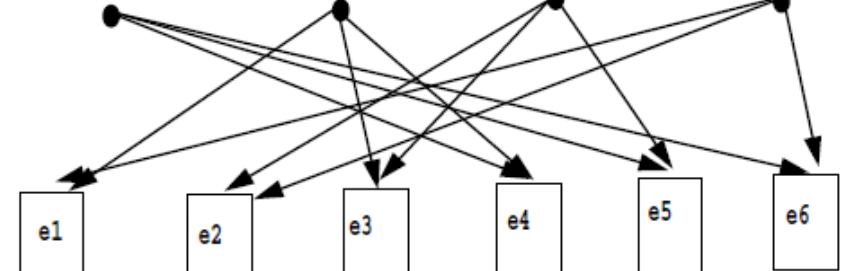
Beispiel Tetraeder



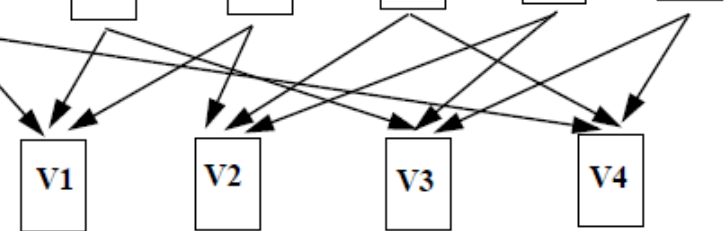
Loops:



Edges:



Vertices:



Topological Information

Vertex assignment
(Geometry)

(x_1, y_1, z_1)

(x_2, y_2, z_2)

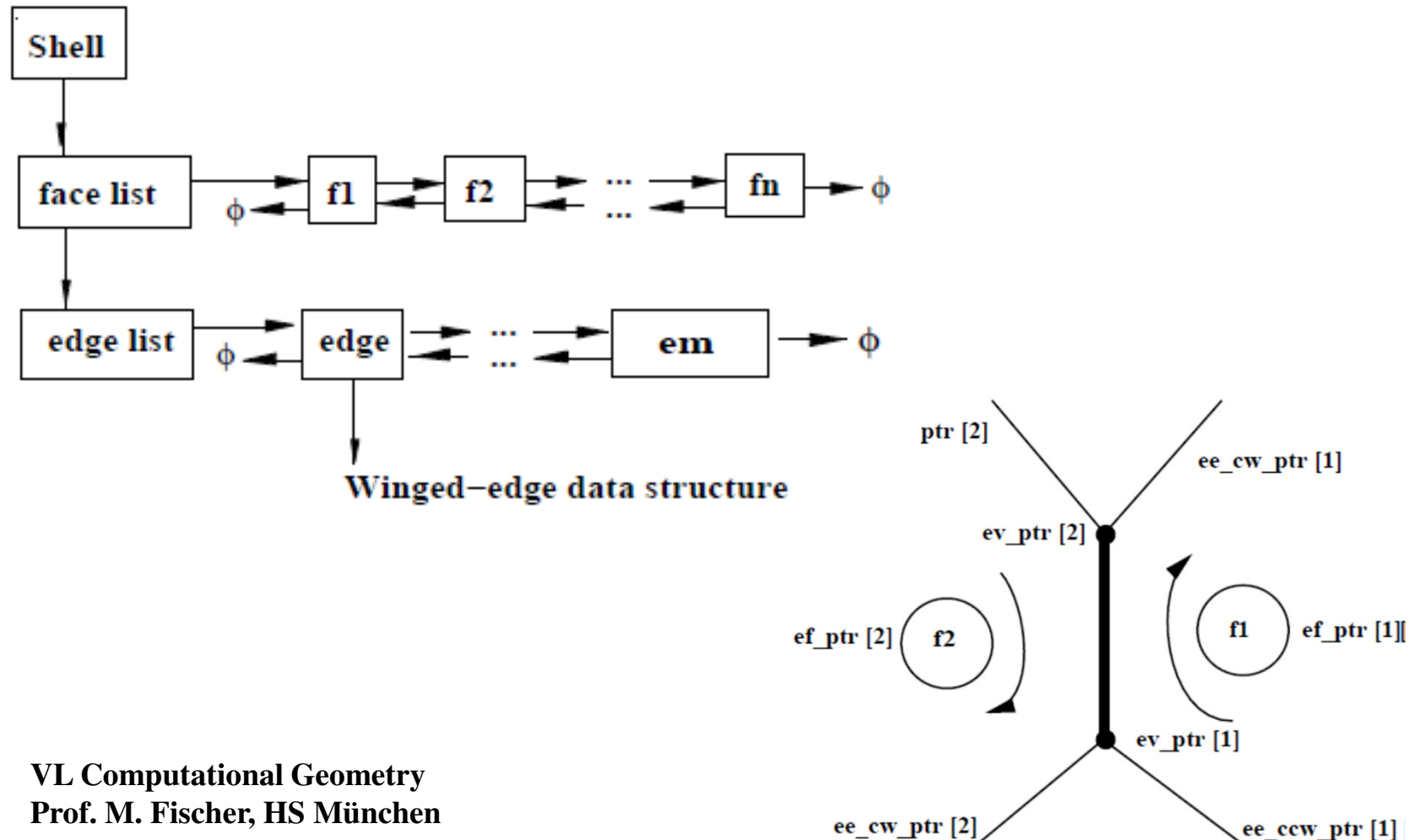
(x_3, y_3, z_3)

(x_4, y_4, z_4)

Modellierung, 3D

Winged Edge Datenstruktur

Vollständige Information über eine Kante

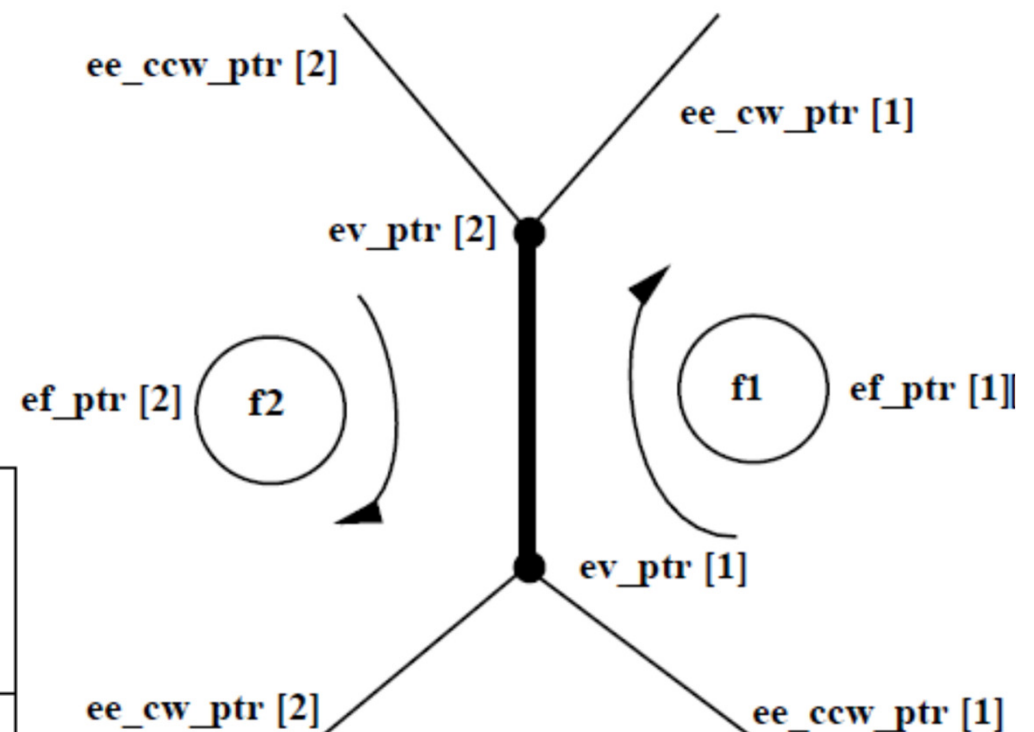


Modellierung, 3D

Winged Edge Datenstruktur

Vollständige Information über eine Kante

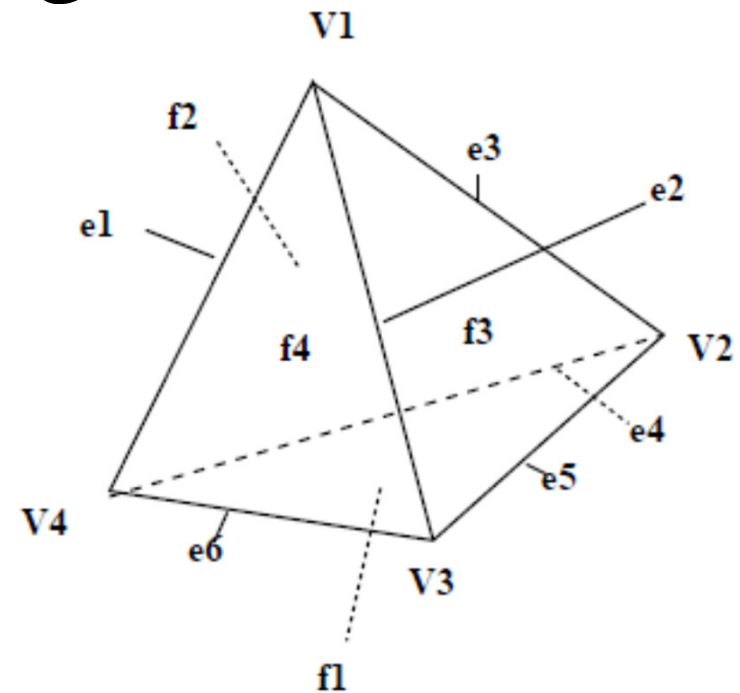
ev_ptr [1]	ev_ptr [2]
ee_cw_ptr [1]	ee_cw_ptr [2]
ee_ccw_ptr [1]	ee_ccw_ptr [2]
ef_ptr [1]	ef_ptr [2]
e_ptr (edge attribute)	



Modellierung, 3D

Winged Edge Datenstruktur

Beispiel Tetraeder



edge	$V[1]$	$V[2]$	$f[1]$	$f[2]$	$CW[1]$	$CCW[1]$	$CW[2]$	$CCW[2]$
e_1	V_1	V_4	f_2	f_4	e_4	e_3	e_2	e_6
e_2	V_3	V_1	f_3	f_4	e_3	e_5	e_6	e_1
e_3	V_1	V_2	f_3	f_2	e_5	e_2	e_1	e_4
e_4	V_2	V_4	f_1	f_2	e_6	e_5	e_3	e_1
e_5	V_2	V_3	f_3	f_1	e_2	e_3	e_4	e_6
e_6	V_4	V_3	f_1	f_4	e_5	e_4	e_1	e_2