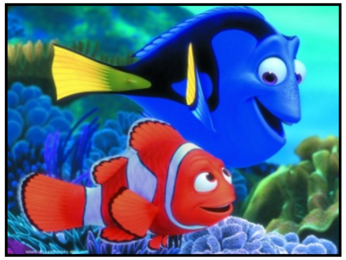
2. Geometrische Modellierung

2.5 Weitere Modellierungsansätze in der Computergraphik

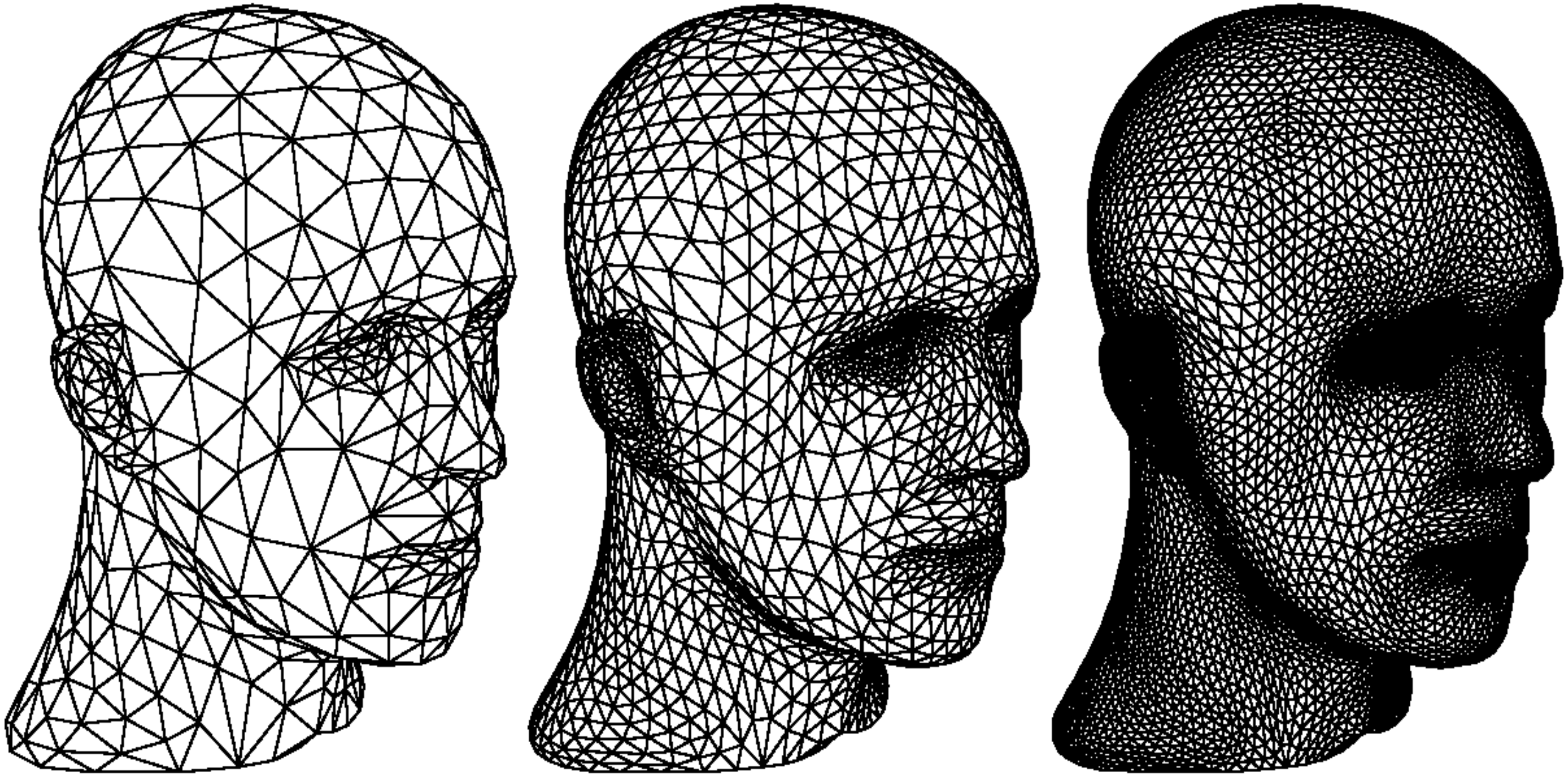
- Die bisher eingeführten Modellierungsansätze in der Computergraphik sind zwar die häufigsten, jedoch nicht die einzigen
- Für bestimmte Anwendungen können spezielle Modellierungstechniken genutzt werden
  - Verarbeitung sehr großer Datenmengen,
  - Verarbeitung spezifischer Objekte
    (z.B. natürliche Objekte und Phänomene, medizinische Daten).

- Deshalb werden spezielle Modelle verwendet:
  - **Unterteilungsflächen (Subdivision surfaces)**
  - **Punktbasierte Modelle**
  - **Bildbasierte Modelle**
  - **Volumenmodelle / implizite Modelle**
  - **Multiresolution Modelle**

- **Unterteilungsflächen (Subdivision surfaces)**
  - Idee: Grobes Polygonnetz + Unterteilungsregeln
  - Wiederholtes Unterteilen führt zu glatter Fläche
  - Standard für Animationsfilme



- **Unterteilungsflächen (Subdivision surfaces)**
  - Idee: Grobes Polygonnetz + Unterteilungsregeln
  - Wiederholtes Unterteilen führt zu glatter Fläche

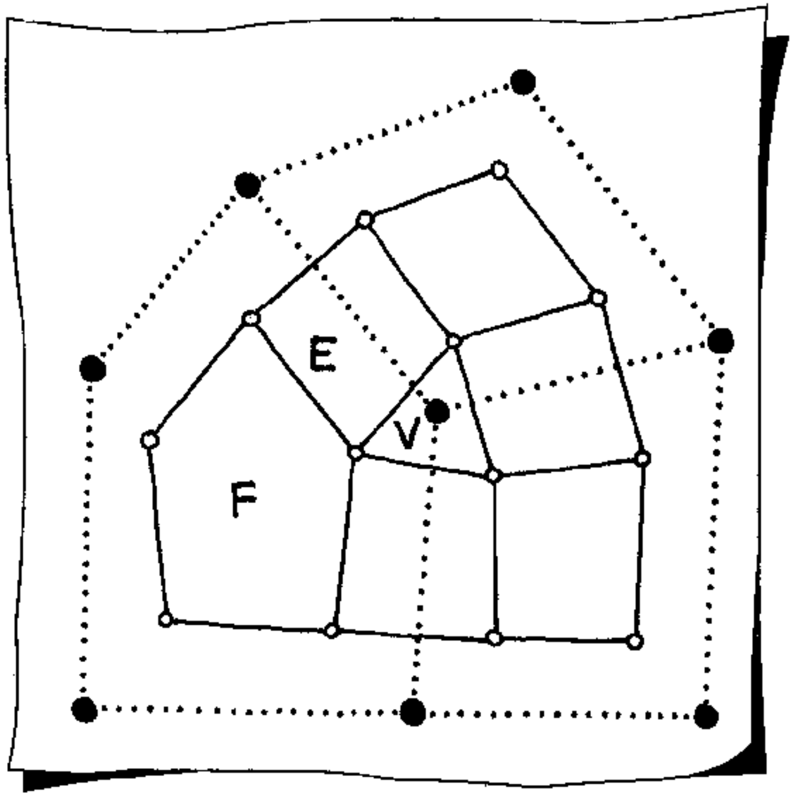


- **Unterteilungsflächen (Subdivision surfaces)**
  - Beispiel: Doo-Sabin Interpolant

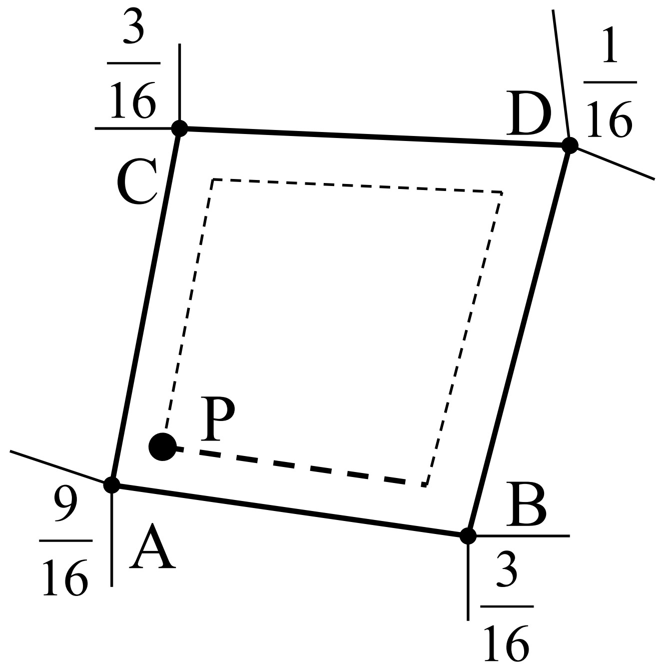
- **Unterteilungsflächen (Subdivision surfaces)**
  - Beispiel: Doo-Sabin Interpolant

Input: Polygonnetz
Algorithmus:
1) Bilde neue Punkte in jedem Face
2) Verbinde diese Punkte, um neue Faces zu erhalten: F-faces, E-faces, V-faces

Repeat ...
Output: Polygonnetz;
Grösstenteils Vierecks-faces
mit Ausnahme von F- & V-faces;
valence = 4 überall

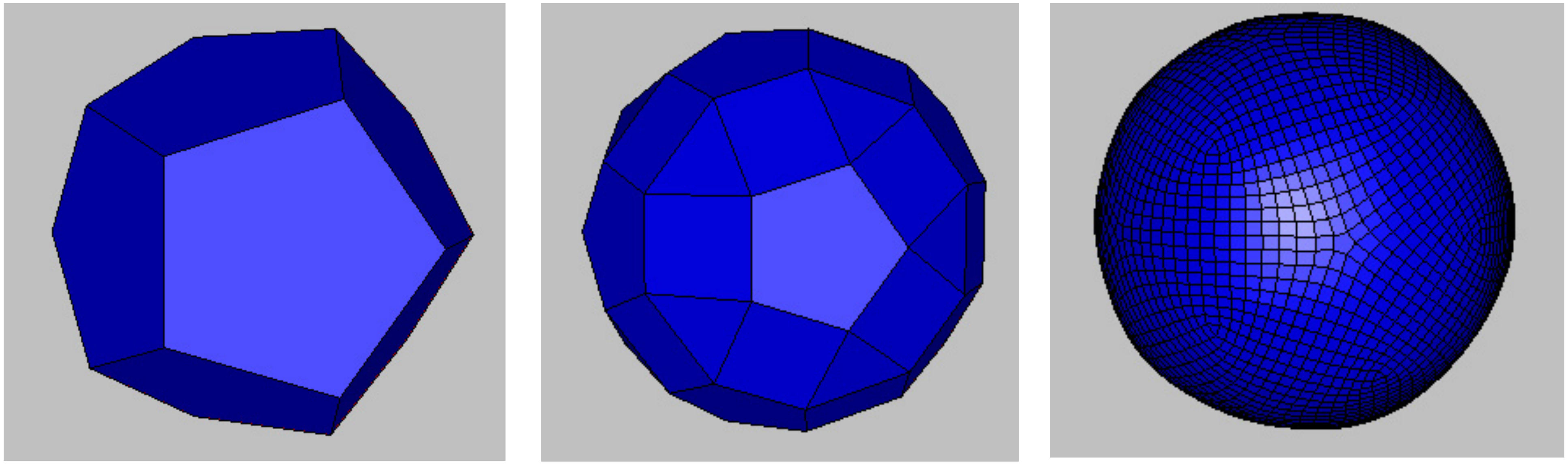


- **Unterteilungsflächen (Subdivision surfaces)**
  - Beispiel: Doo-Sabin Interpolant
  - *Für Viereck:* $P = (9/16)A + (3/16)B + (3/16)C + (1/16)D$

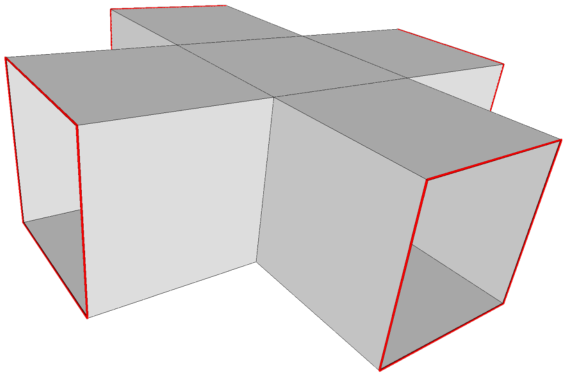


- Damit wird ein alter vertex durch 4 neue ersetzt
- Im Limit ist die Fläche biquadratisch und C1 stetig überall.

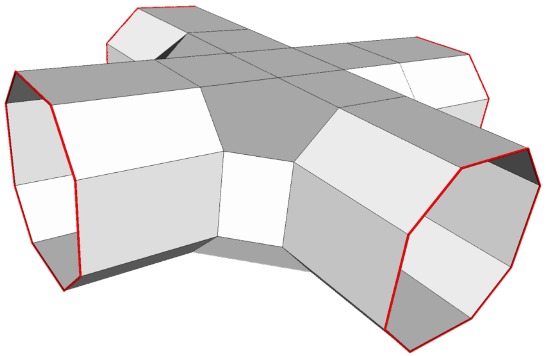
- **Unterteilungsflächen (Subdivision surfaces)**
  - Beispiel: Doo-Sabin Interpolant



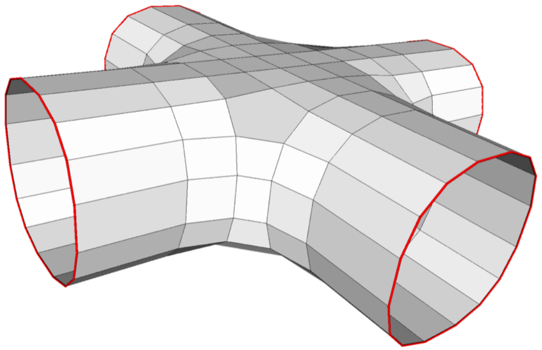
- **Unterteilungsflächen (Subdivision surfaces)**
  - Beispiel: Doo-Sabin Interpolant



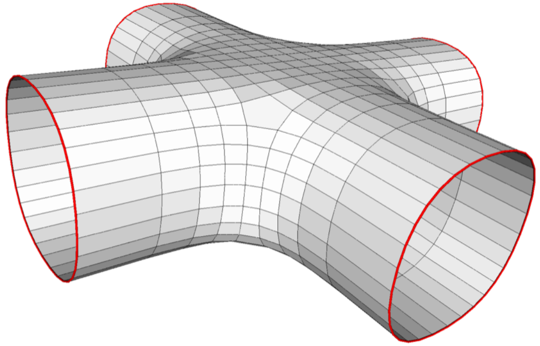
(0) 18 faces



(1) 54 faces



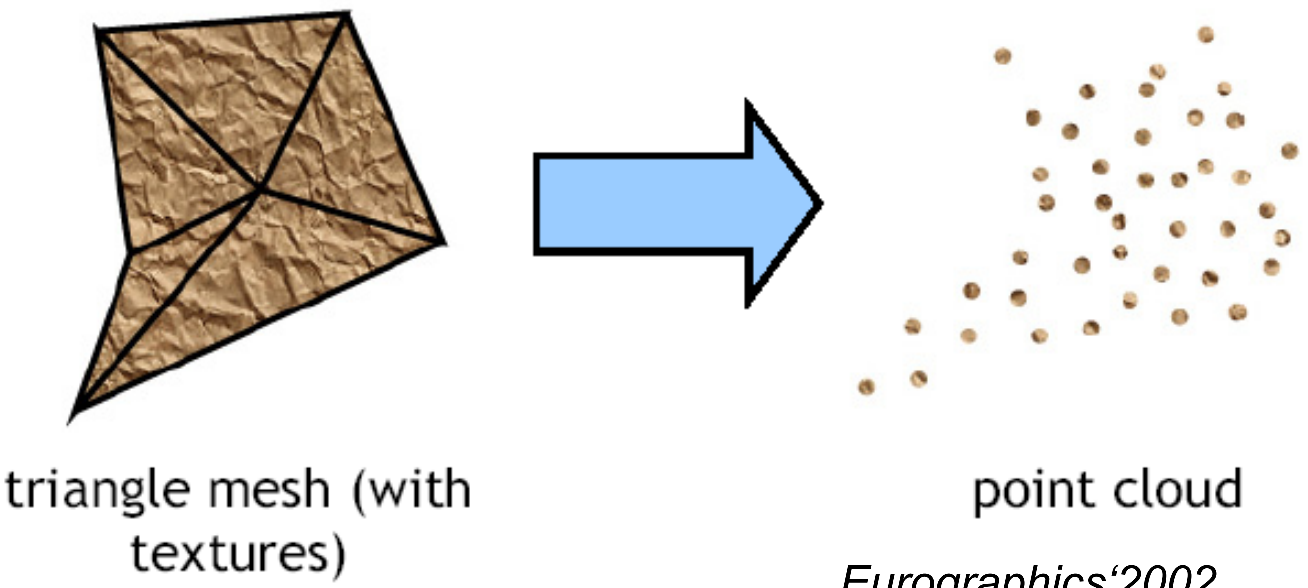
(2) 190 faces



(3) 702 faces

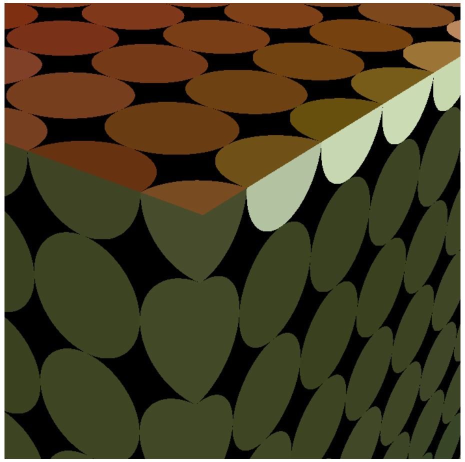
- **Unterteilungsflächen (Subdivision surfaces)**
  - Es gibt viele weitere Schemen für Unterteilungsflächen, z.B.
    - Catmull-Clark
    - Loop
    - Sqrt(3)
    - Butterfly
    - ...

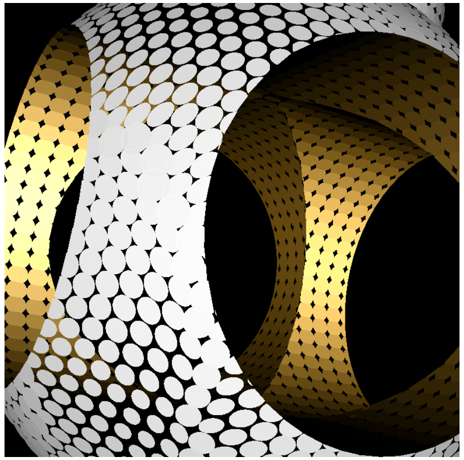
- **Punktbasierte Modelle**
  - Objekte werden anhand von Punkten beschrieben. Die Anzahl der verwendeten Punkte ist ein Maß für die Genauigkeit ihrer Repräsentation.



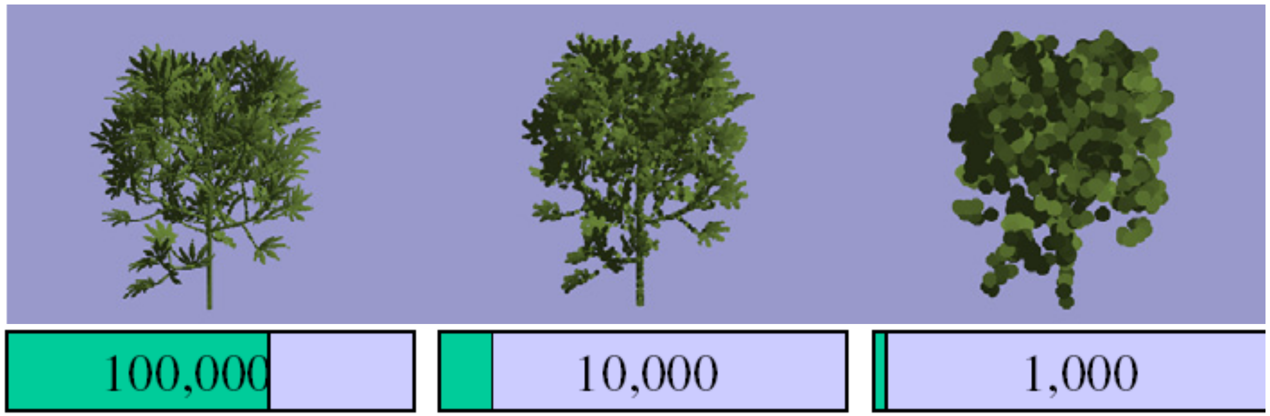
Eurographics'2002, Course Notes

- **Punktbasierte Modelle**





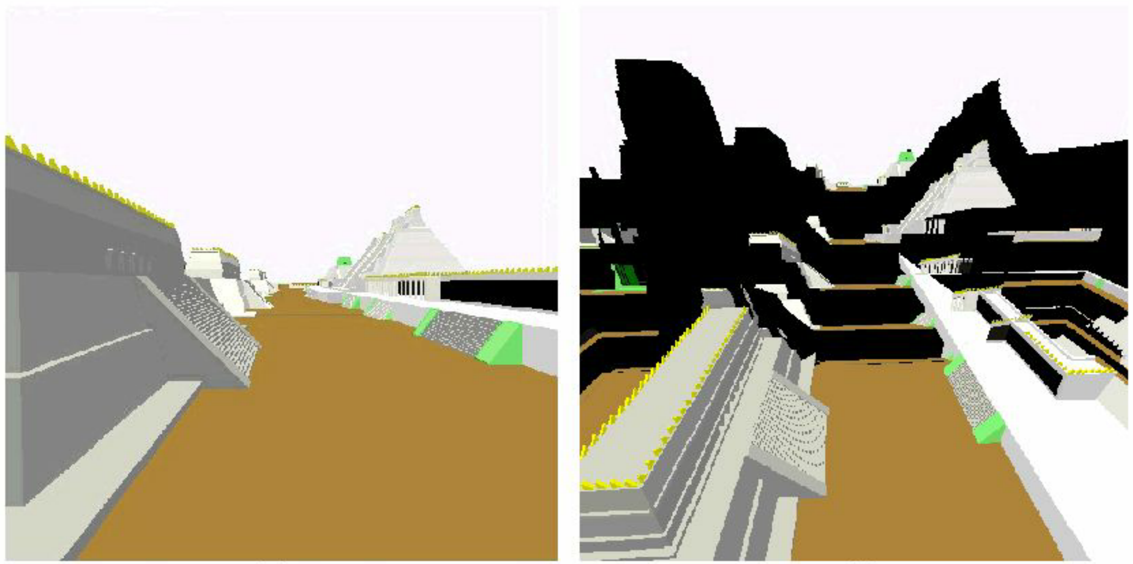
Repräsentation durch splats



- **Punktbasierte Modelle**

- **Bildbasierte Modelle**
  - 3D-Szenen werden mit Hilfe von Bildbeschreibungen modelliert.

    Hierfür gibt es verschiedene Ansätze: Bei Verwendung der Imposter-Technik z.B. werden Bilder und geometrische Objekte gleichzeitig verwendet.



(a) (b)

Beispiel zum Einsatz von Impostern, aus Jeschke, 2001

a) gerendertes Bild aus Betrachterposition
b) Vogelperspektive

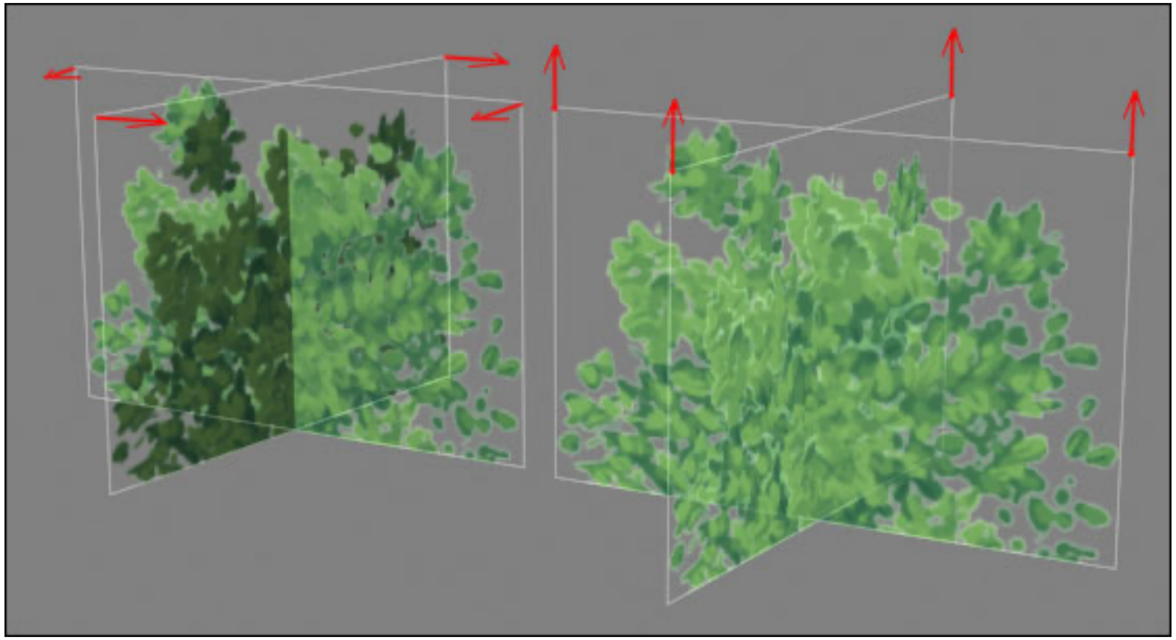
- **Bildbasierte Modelle**
  - 3D-Szenen werden mit Hilfe von Bildbeschreibungen modelliert.

    Hierfür gibt es verschiedene Ansätze: Bei Verwendung der Imposter-Technik z.B. werden Bilder und geometrische Objekte gleichzeitig verwendet.

- **Bildbasierte Modelle**
  - Spezialfall für imposter: billboards für Baum-Modelle

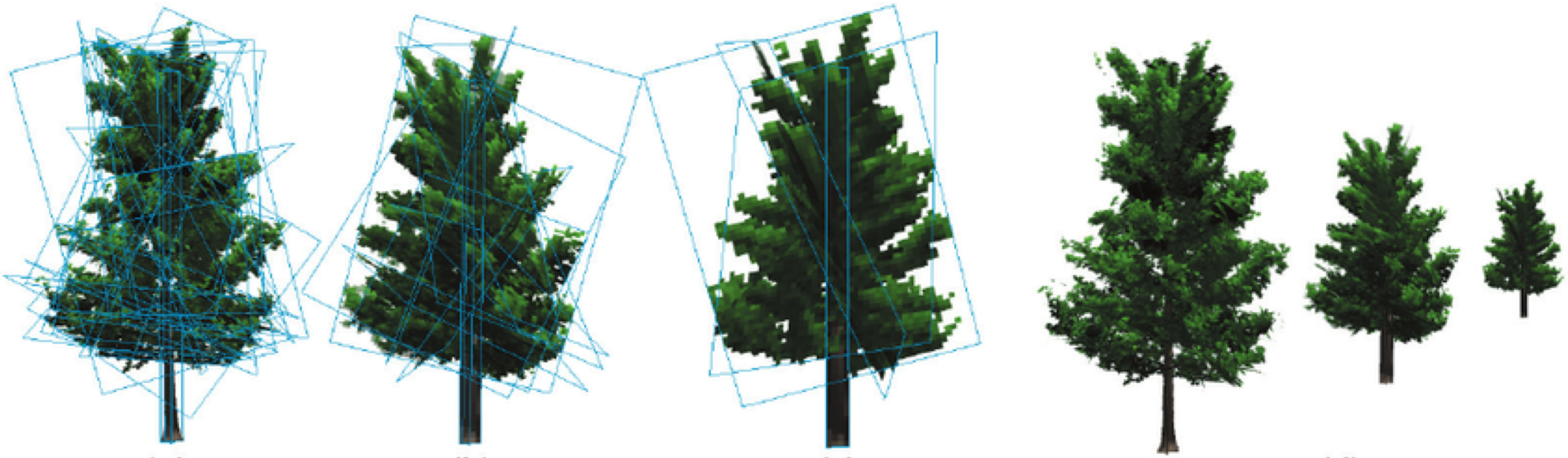
https://developer.x-plane.com/article/building-custom-3-d-trees/

- **Bildbasierte Modelle**
  - Spezialfall für imposter: billboards für Baum-Modelle



https://blender.stackexchange.com/questions/265790/best-way-to-create-tree-billboard

- **Bildbasierte Modelle**
  - Spezialfall für imposter: billboards für Baum-Modelle

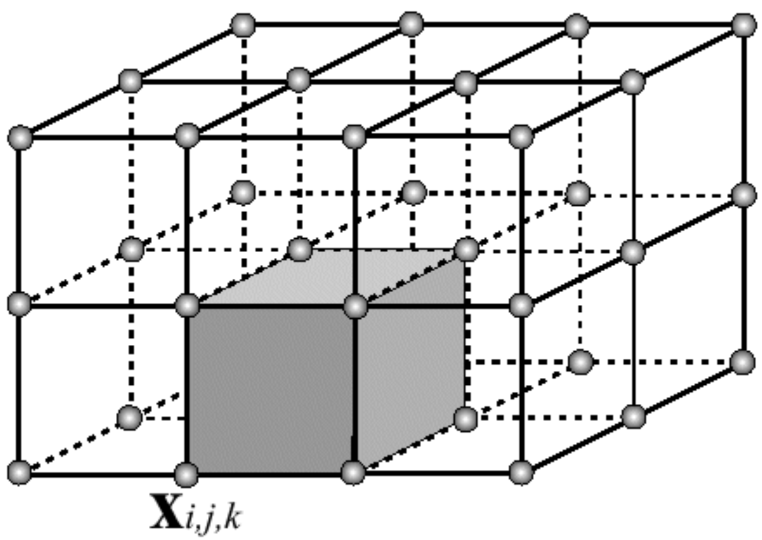


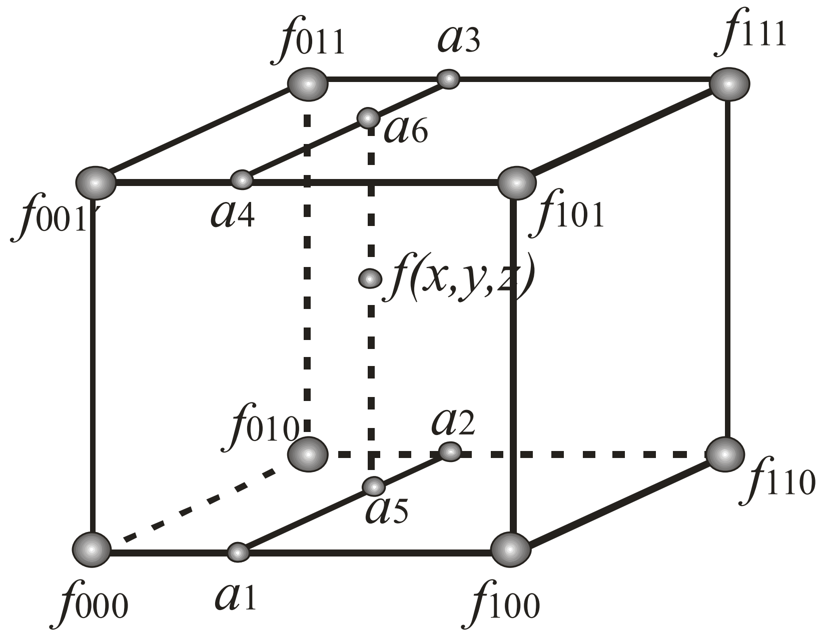
(a) (b) (c) (d)

https://www.researchgate.net/figure/Levels-of-detail-for-a-Billboard-Cloud-tree-a-24-planes-b-11-planes-c-4-planes_fig6_221314842

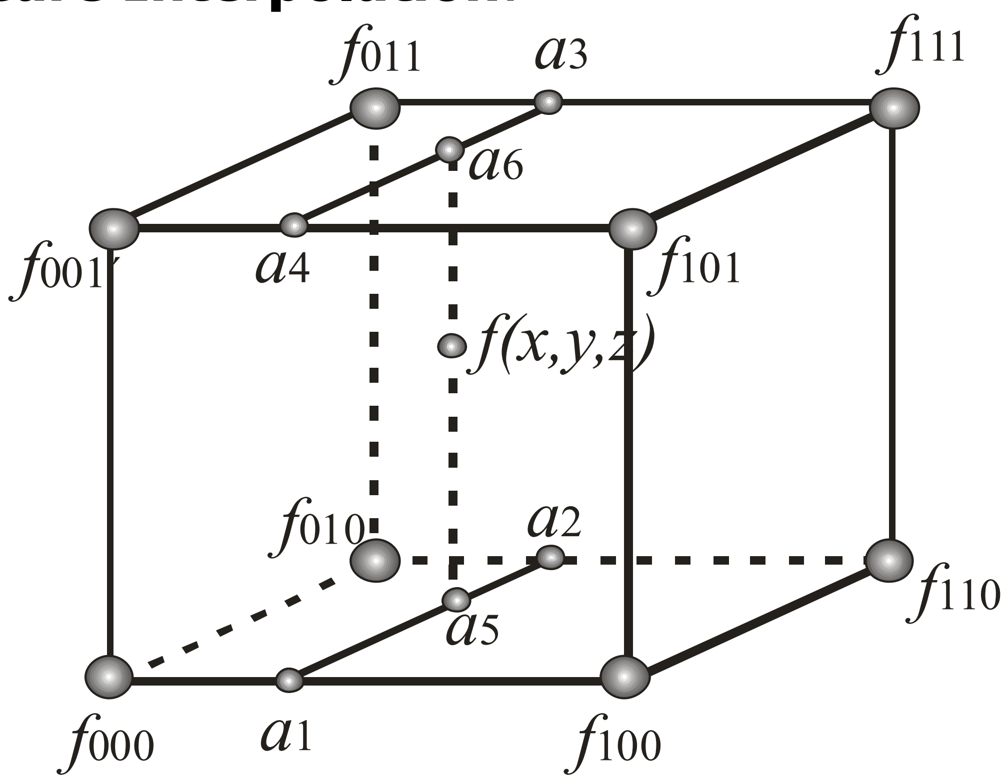
- **Volumenmodelle/implizite Modelle**
  - Modell als Isofläche eines skalaren Feldes gegeben
  - skalares Feld: gegeben durch reguläres Gitter
    - skalarer Wert an Gitterpunkten
    - trilineare Interpolation in jeder Gitterzelle

- **Volumenmodelle/implizite Modelle**
  - Modell als Isofläche eines skalaren Feldes gegeben
  - skalares Feld: gegeben durch reguläres Gitter
    - skalarer Wert an Gitterpunkten
    - trilineare Interpolation in jeder Gitterzelle





Sei $f_{ijk} = f(i,j,k)$ for $i,j,k \in \{0,1\}$. Dann kann der Wert $f(x,y,z)$ für einen Punkt im Inneren der Zelle mit den lokalen Koordinaten $(x,y,z) \in [0,1]^3$ berechnet werden durch **trilineare Interpolation**:

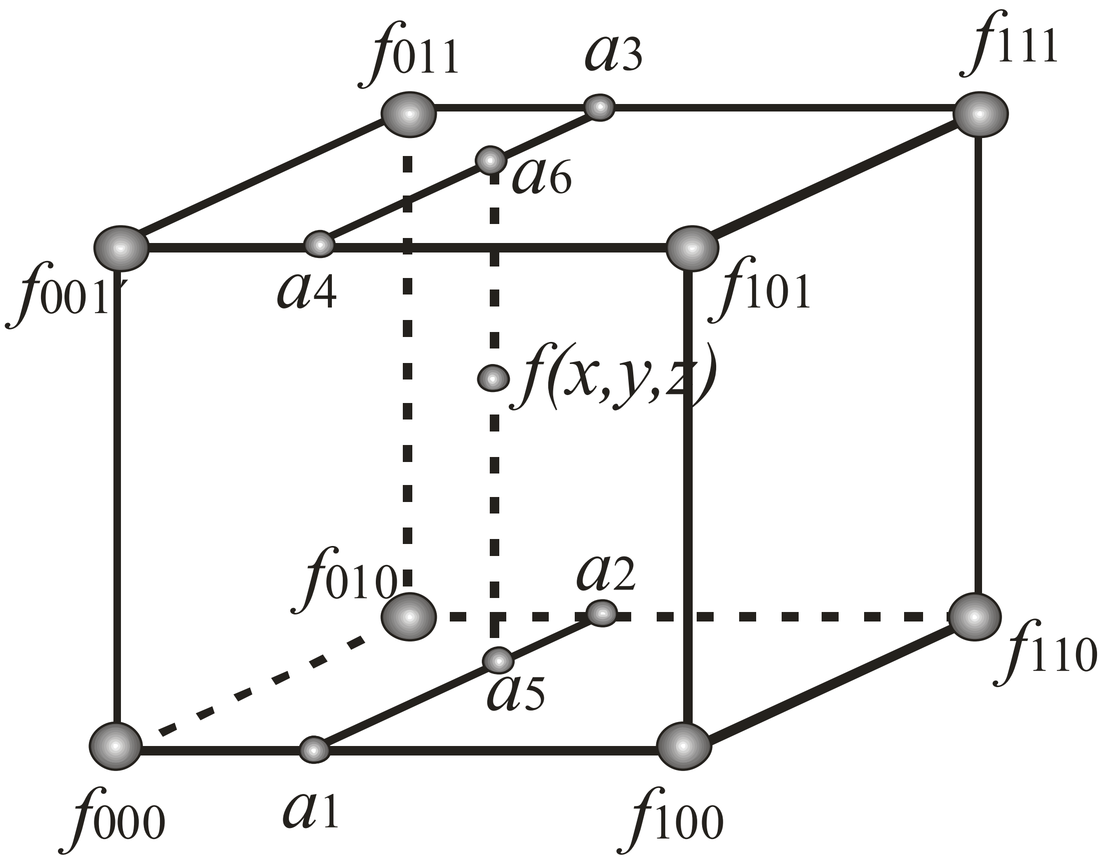


$$a_1 = (1-x)*f_{000} + x*f_{100}$$
$$a_2 = (1-x)*f_{010} + x*f_{110}$$
$$a_3 = (1-x)*f_{011} + x*f_{111}$$
$$a_4 = (1-x)*f_{001} + x*f_{101}$$
$$a_5 = (1-y)*a_1 \quad + y*a_2$$
$$a_6 = (1-y)*a_4 \quad + y*a_3$$

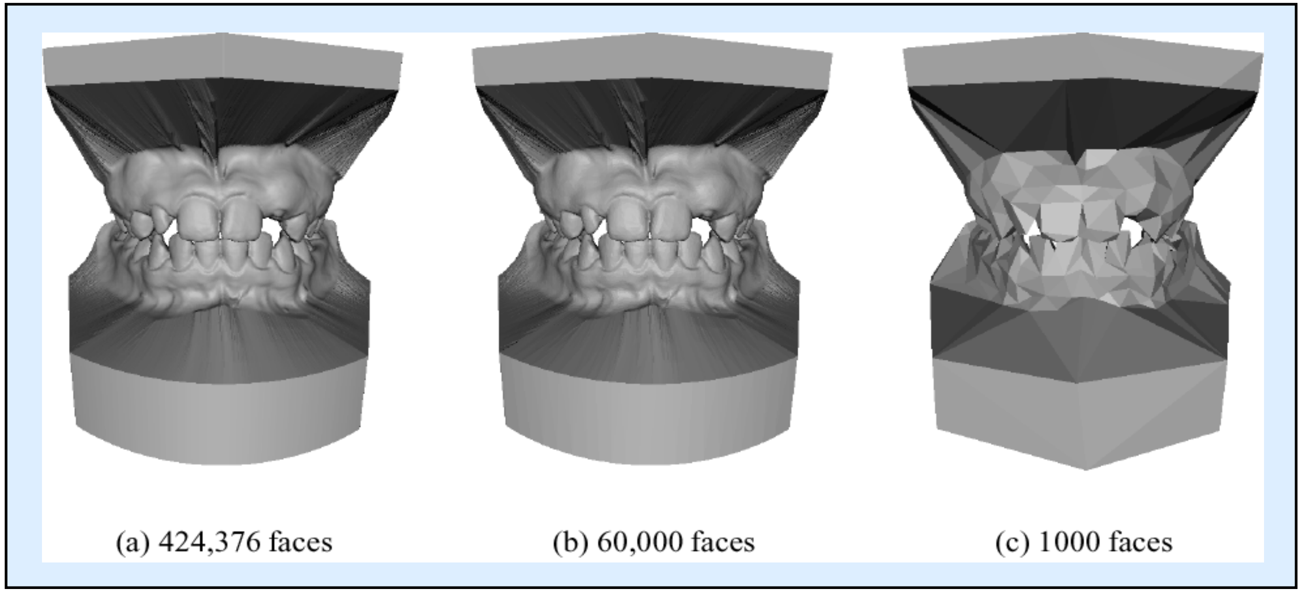
$$f(x,y,z) = (1-z)*a_5 \quad + z*a_6$$

Trinlineare Interpolation anders aufgeschrieben:

$$
\begin{aligned}
f(x,y,z) = \\
& (1-x)*(1-y)*(1-z)*f_{000} \\
+ & \quad x*(1-y)*(1-z)*f_{100} \\
+ & \quad x*\quad y*(1-z)*f_{110} \\
+ & (1-x)*\quad y*(1-z)*f_{010} \\
+ & (1-x)*(1-y)*\quad z*f_{001} \\
+ & \quad x*(1-y)*\quad z*f_{101} \\
+ & \quad x*\quad y*\quad z*f_{111} \\
+ & (1-x)*\quad y*\quad z*f_{011}
\end{aligned}
$$



- **Multi-Resolution Modelle**
  - Objektbeschreibungen werden in unterschiedlichen Genauigkeitsstufen modelliert und im Darstellungsprozess verwendet

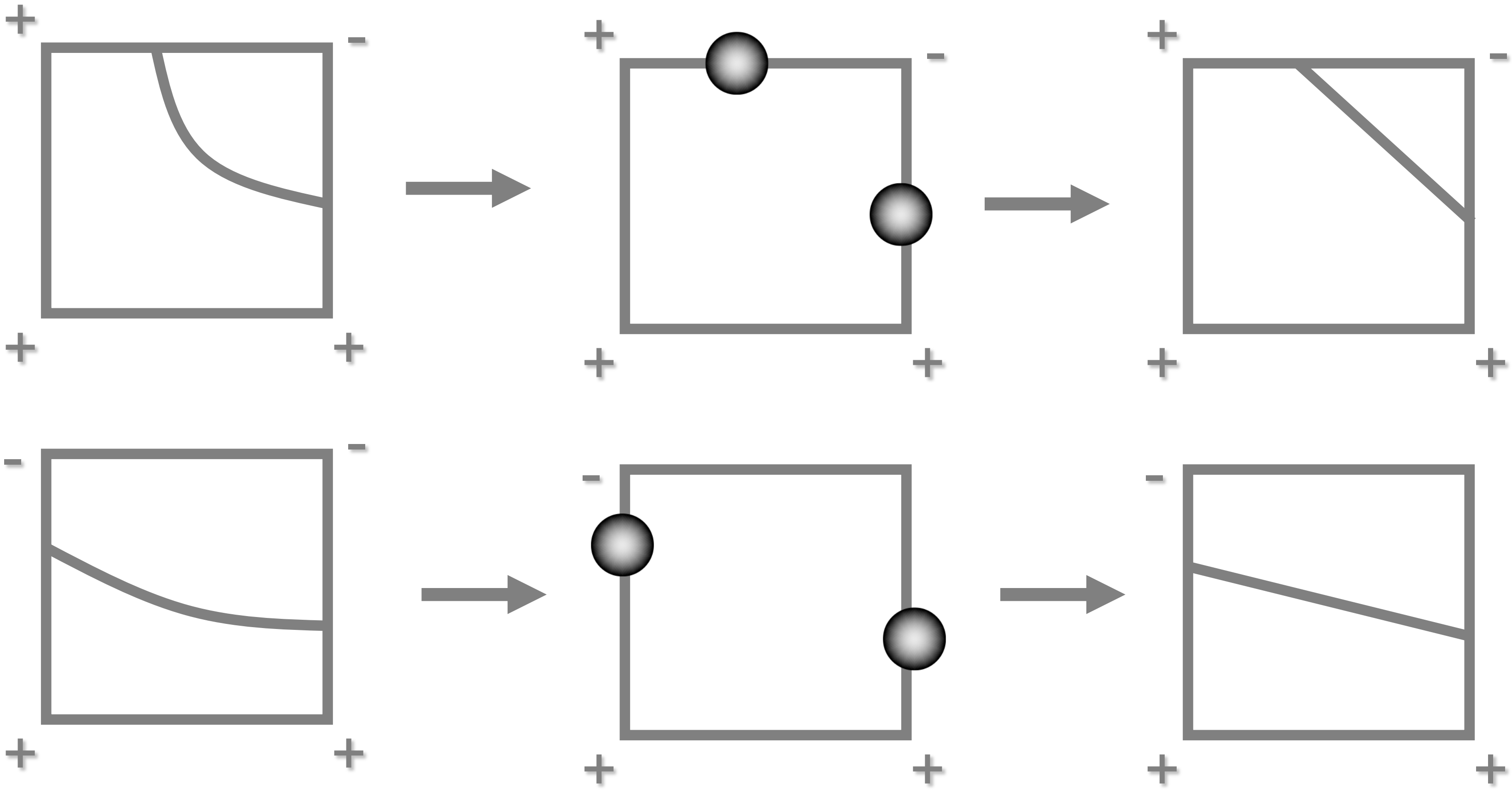


Gescanntes Polygonmodell eines Gebisses in 3 Genauigkeitsstufen aus Garland ´99
a) Original, b) Approximation (14% der Polygone des Originals), c) grobe Approximation

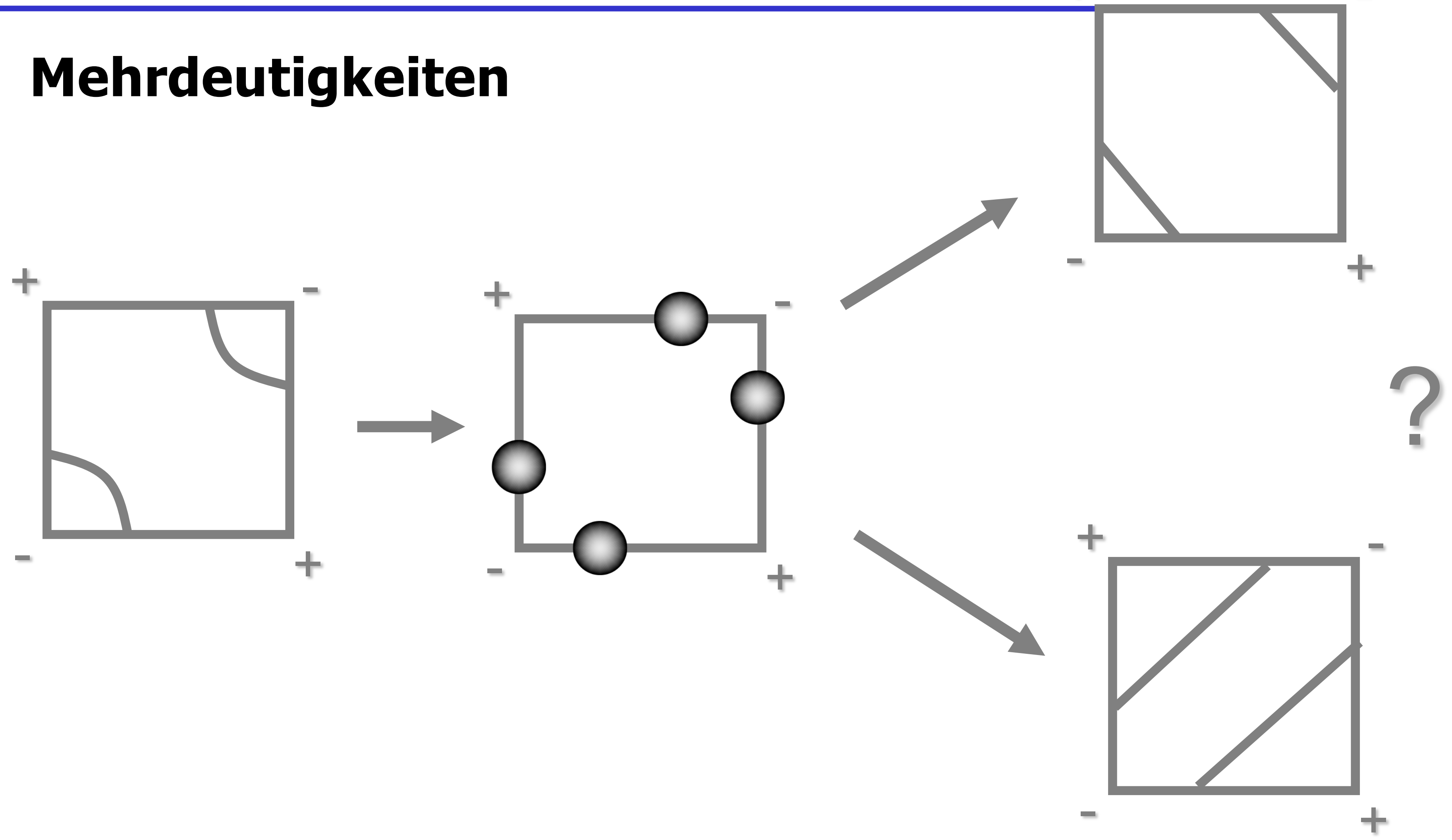
- **2.6. Umwandlung zwischen verschiedenen Modellen**
  - Verschiedene Modelle haben für spezielle Anwendungen Stärken und Schwächen
  - Zwischen verschiedenen Modellen sind Umwandlungen möglich

- **Umwandlung zwischen verschiedenen Modellen:**
  - Volumenmodell -> Dreiecksnetz:
  - 2D: betrachte jede Zelle einzeln

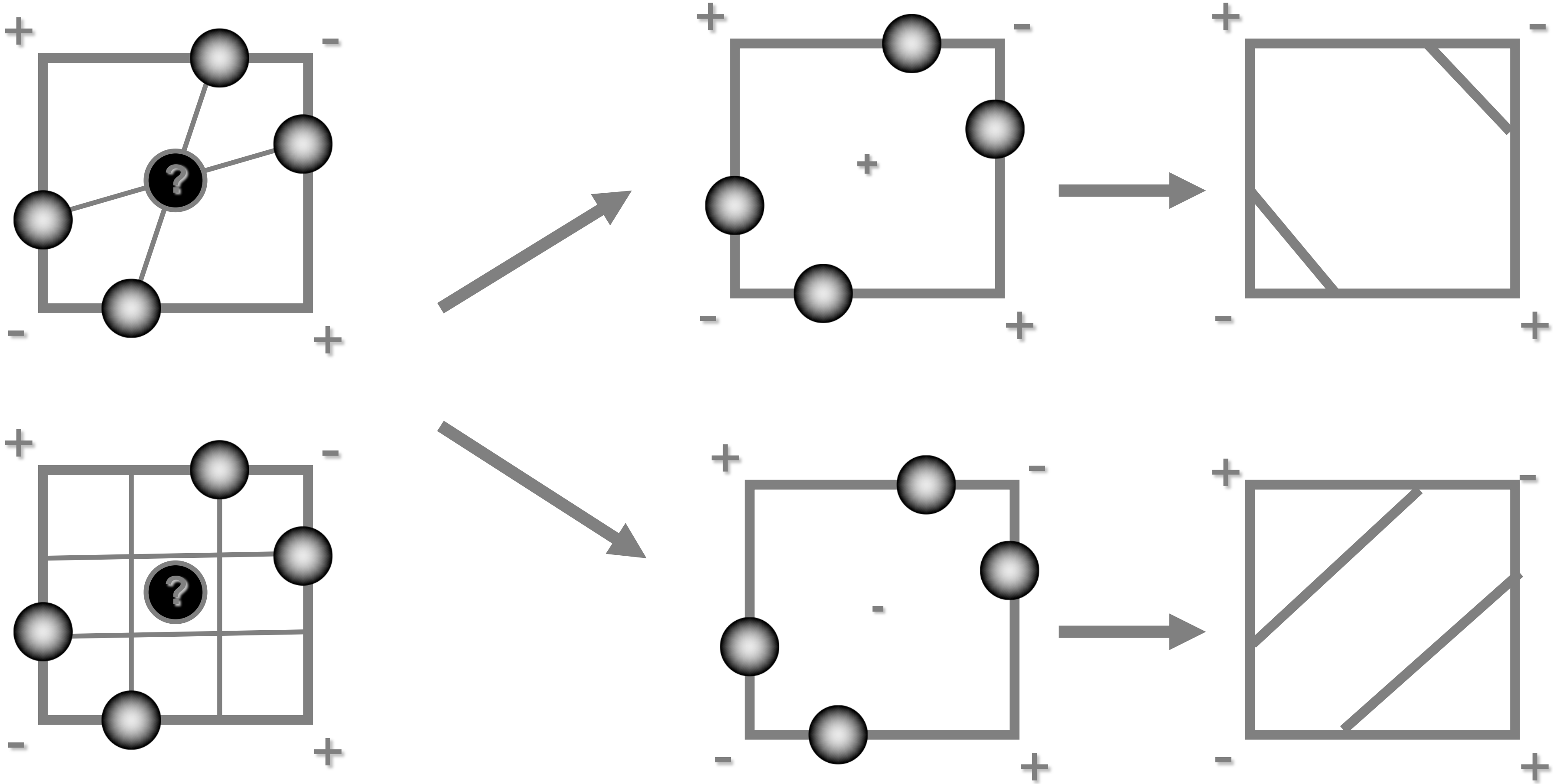
- **Umwandlung zwischen verschiedenen Modellen:**
  - Volumenmodell -> Dreiecksnetz:
  - 2D: betrachte jede Zelle einzeln
  - Skalarwerte in den Eckpunkten gegeben
  - Klassifikation der Eckpunkte
    - \+ Skalarwert > Isowert
    - \- Skalarwert < Isowert
  - Bilineare Interpolation innerhalb der Zelle
  - Finde Schnitte der Isokurve auf allen Kanten mit (+/-)-Konfiguration
  - Lokales Konnektieren innerhalb der Zelle



- **Mehrdeutigkeiten**

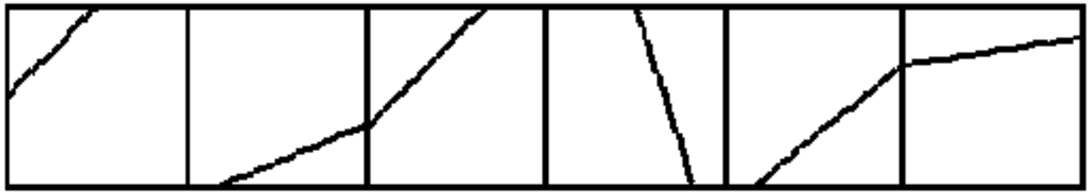


- **Mehrdeutigkeiten lösen:**
- **Prüfe Mittelpunkt durch bilineare Interpolation**

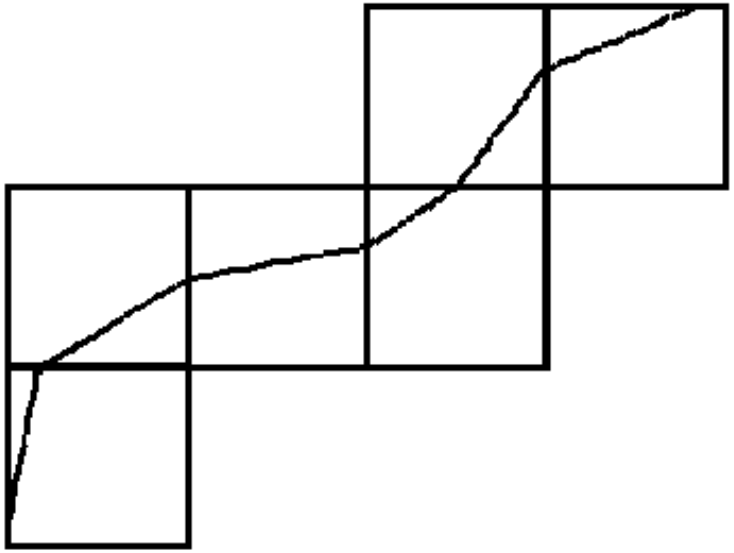


Abarbeitungsreihenfolge

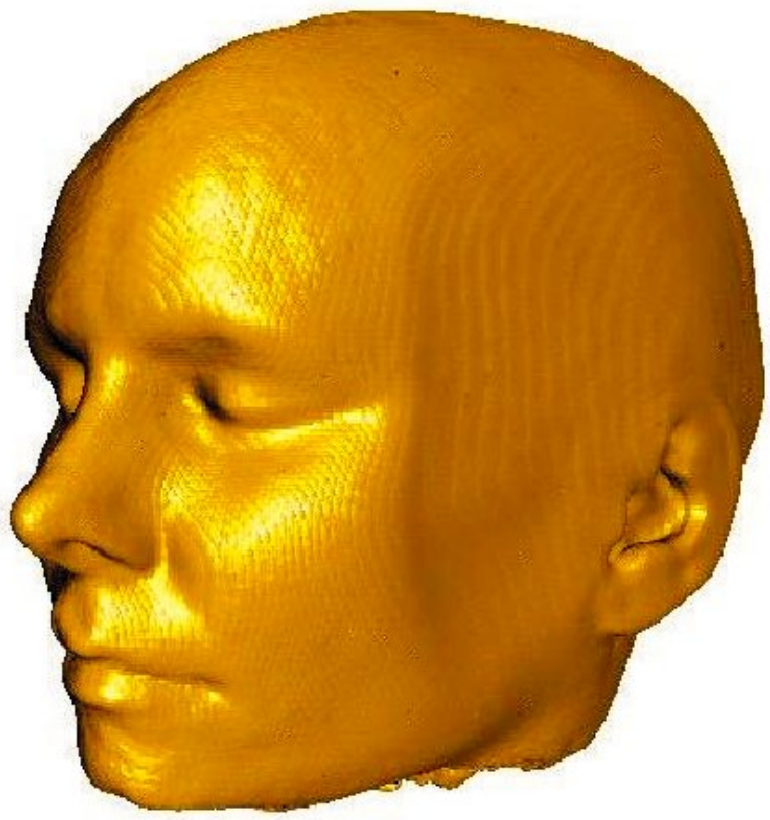
In Gitter-Reihenfolge
(Zeile für Zeile)



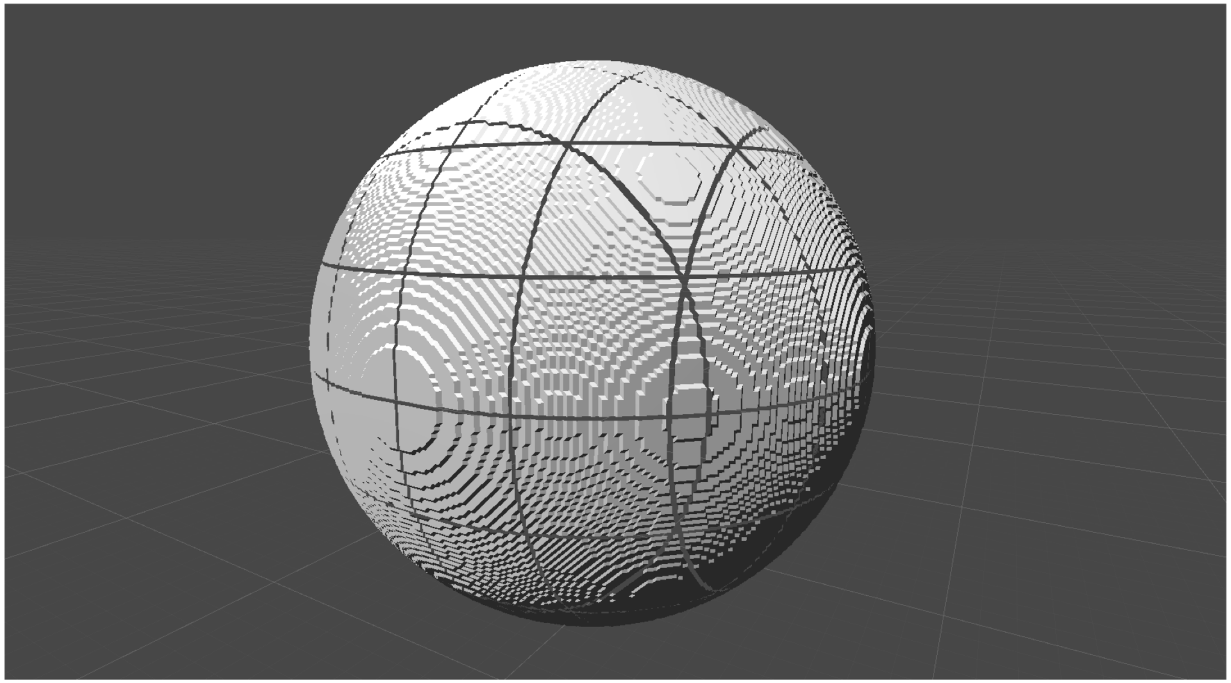
Entlang einer Kurve
(Problem: Startpunkt finden)



- **Umwandlung zwischen verschiedenen Modellen:**
  - Volumenmodell -> Dreiecksnetz:
  - 3D: Marching Cubes



- **Umwandlung zwischen verschiedenen Modellen:**
  - Volumenmodell -> Dreiecksnetz:
  - 3D: binary marching cubes: staircase effects

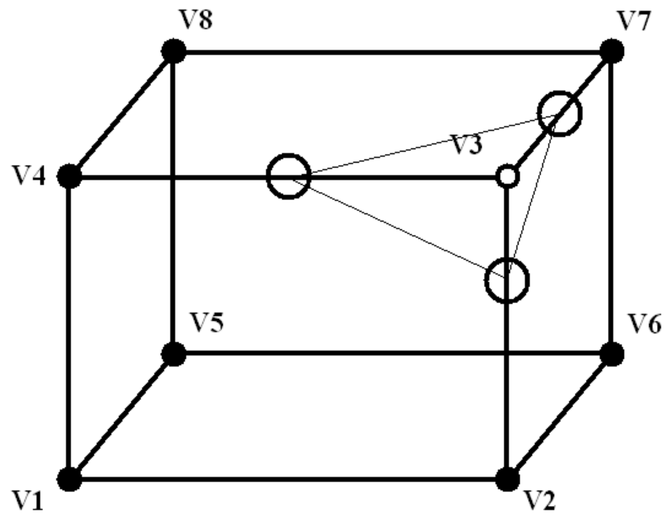


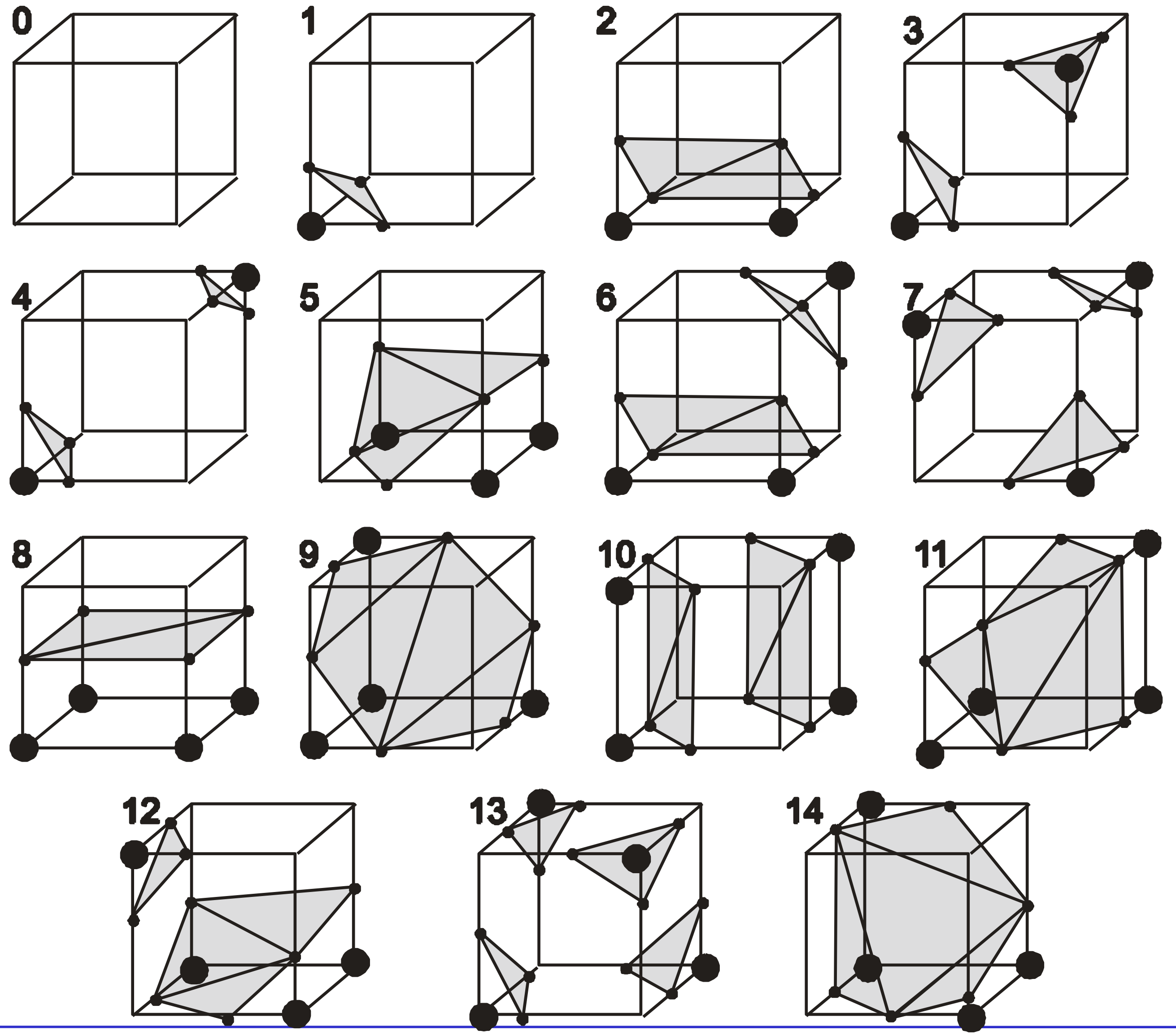
- **Marching Cubes** (Lorensen, 1987)
  - **Ziel**: extrahiere Isofläche aus Volumendateb

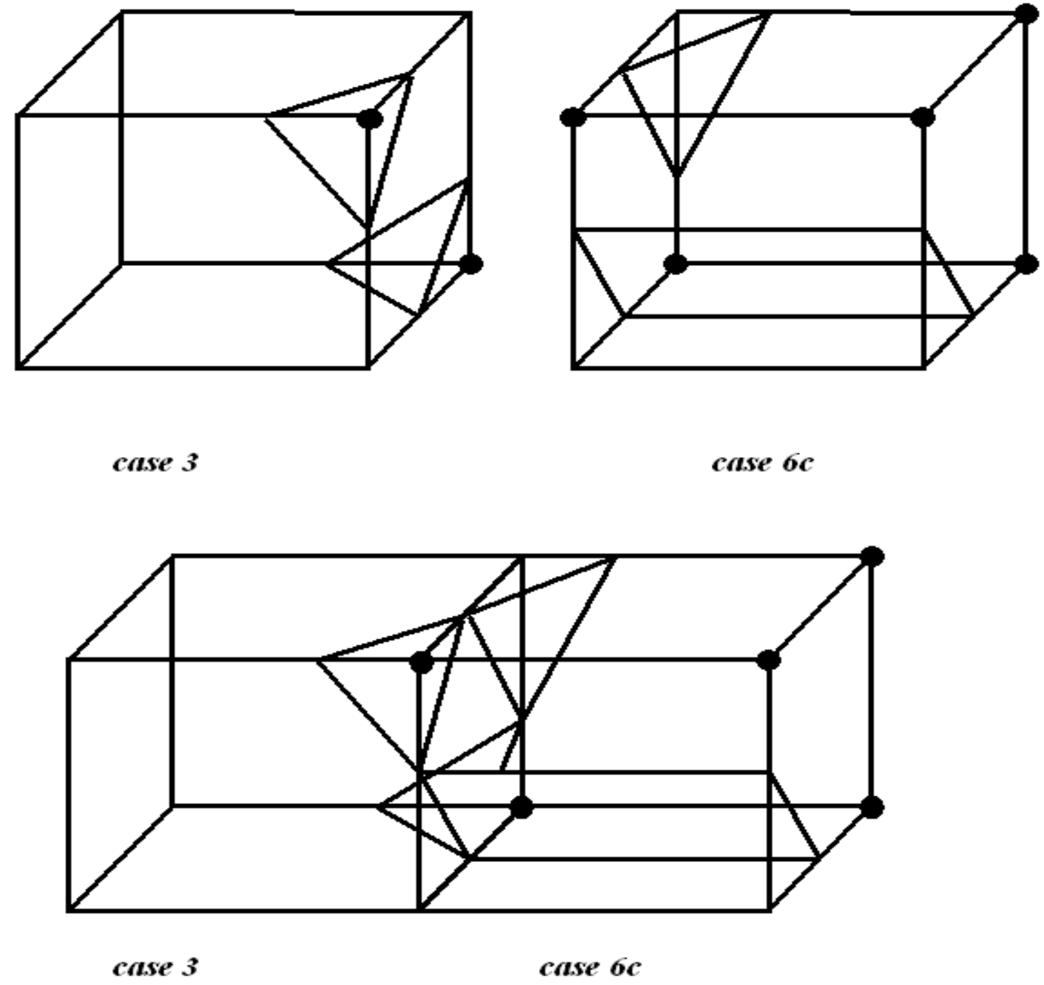
1. Betrachte Zelle mit 8 Datenwerten an den vertices
2. Klassifikation der vertices (inner, outer)
3. Speichere Resultat in index (bit *V8 ... V1*)
4. Erzeuge Kantenliste

   256 Fälle -> Reduktion to 15
5. Interpolation der Dreiecks-vertices

V1	*V2*	*V3*	*V4*	*V5*	*V6*	*V7*	*V8*
0	0	1	0	0	0	0	0

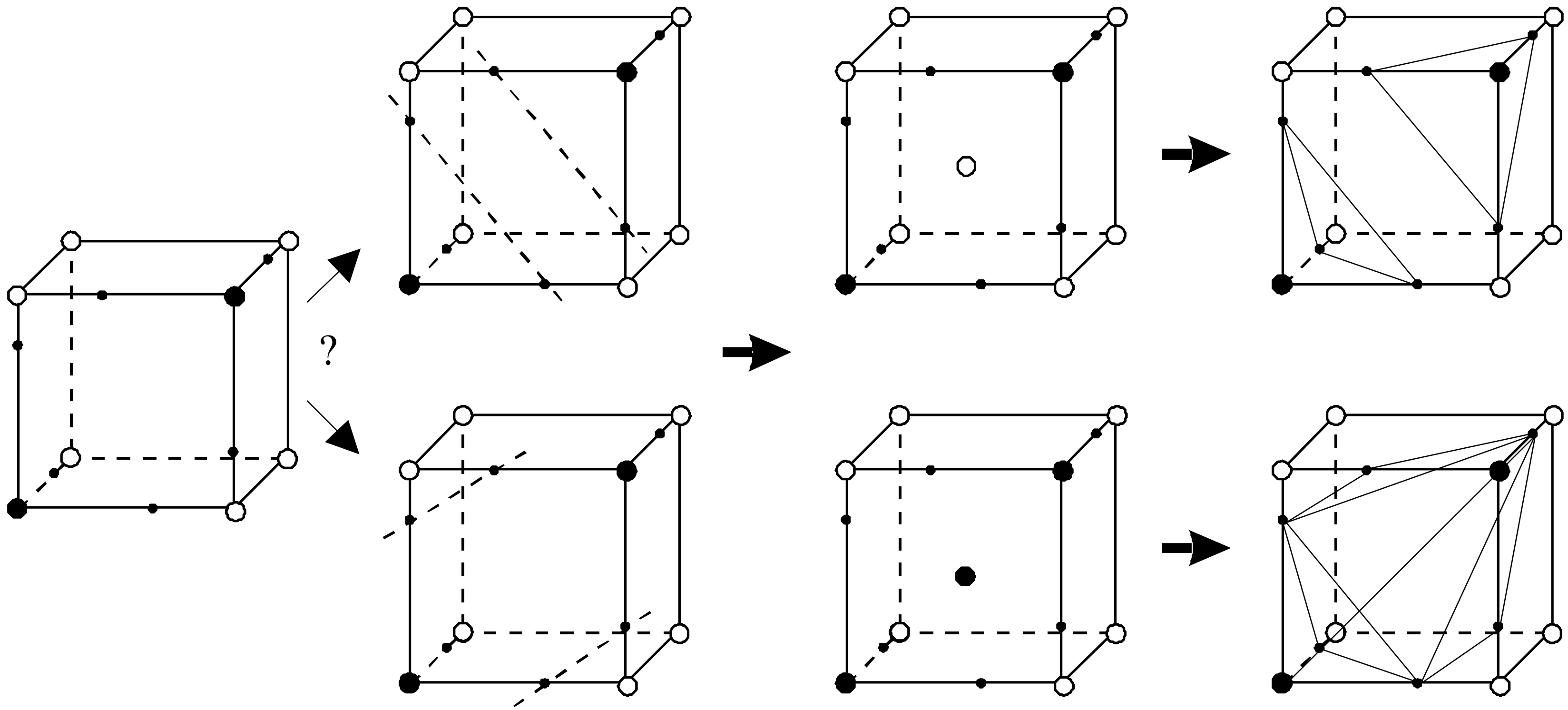



0
1
2
3
4
5
6
7
8
9
10
11
12
13
14



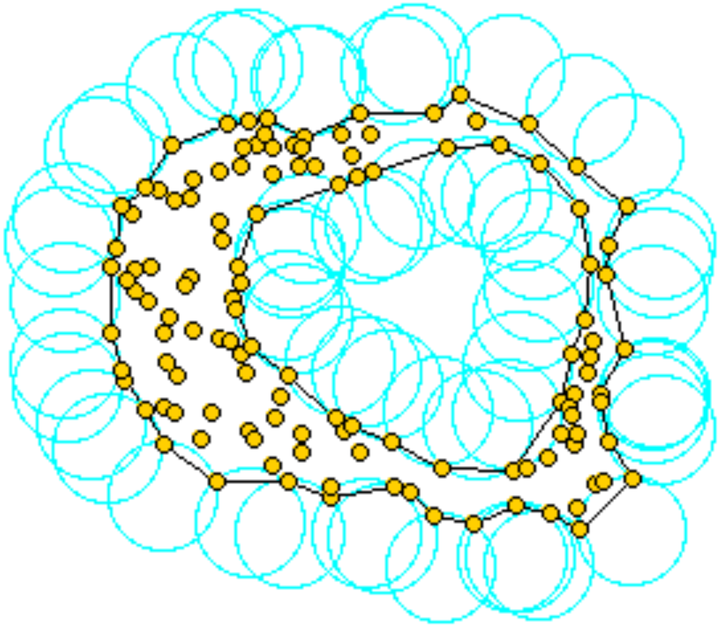
Unwanted holes in Marching Cubes

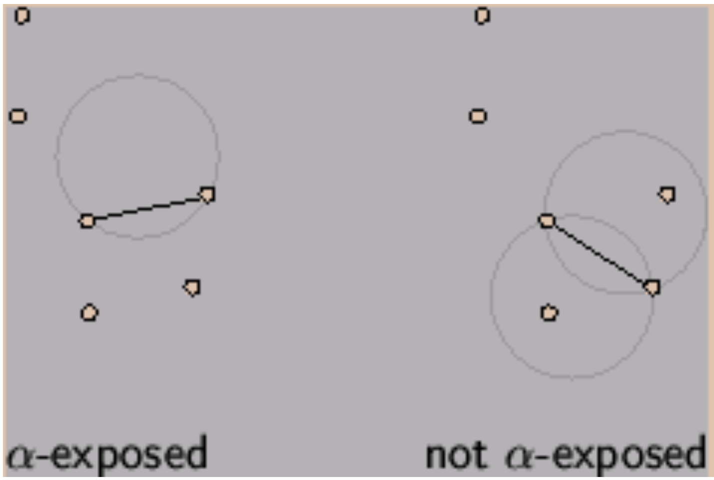
- Mehrdeutigkeiten



- Lösung für jede Fläche (s. 2D-Fall) -> geschlossene Polygone
- diese triangulieren

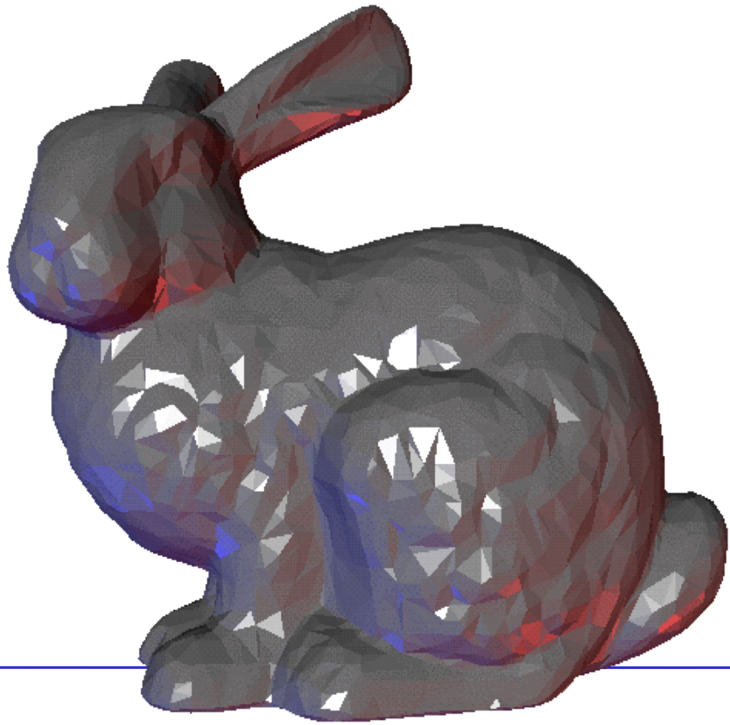
- **Umwandlung zwischen verschiedenen Modellen:**
  - **Punktbasierte Modelle -> Dreiecksnetz:**
  - α-shapes (Edelsbrunner/Mücke)





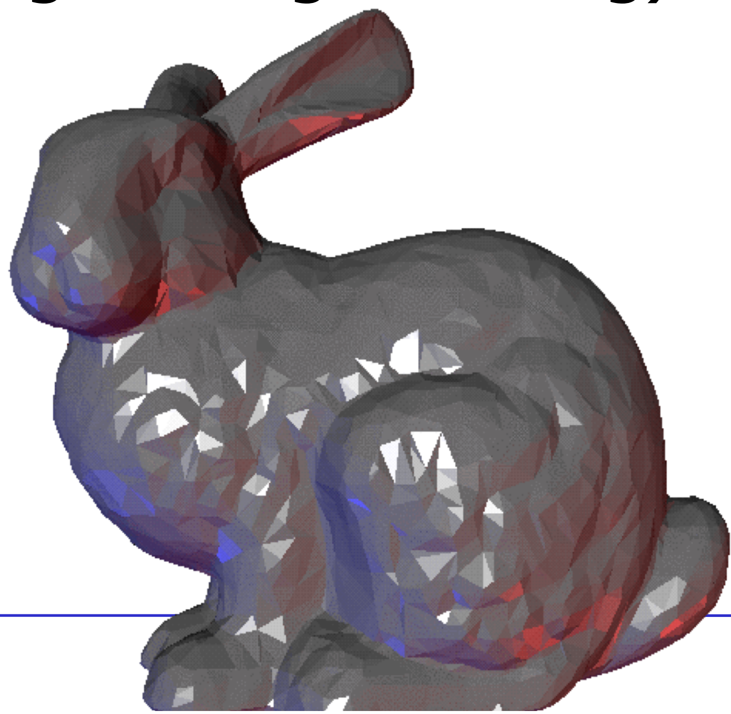
Alpha-shapes

- Trianguliere alle Punkt-tripel, die auf einer Kugel mit dem Radius alpha liegen und in dessen Innerem kein weiterer Punkt der Punktwolke liegt.



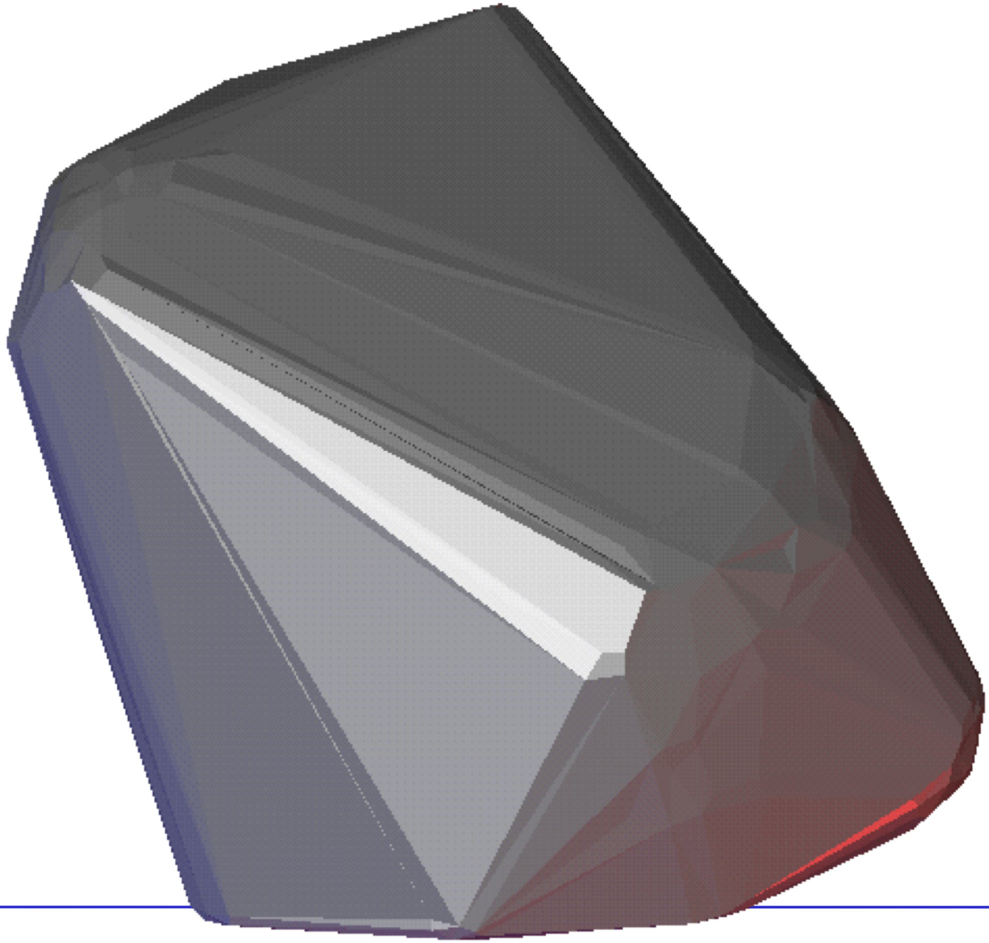
Alpha-shapes

- Beschleunigung: nicht alle Tripel betrachten sondern nur solche die auf der Delaunay-Tetrahedrisierung liegen
- (Delaunay-Triangulierung in Übung)

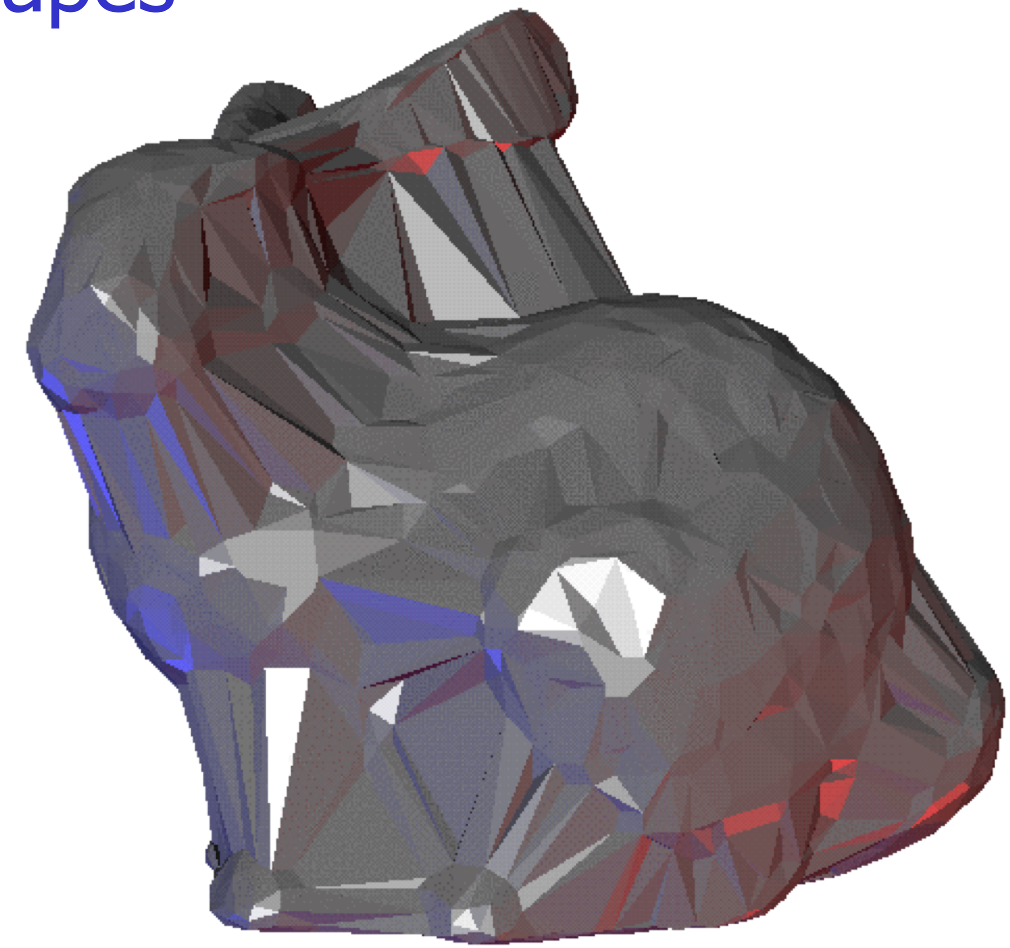


Alpha-shapes

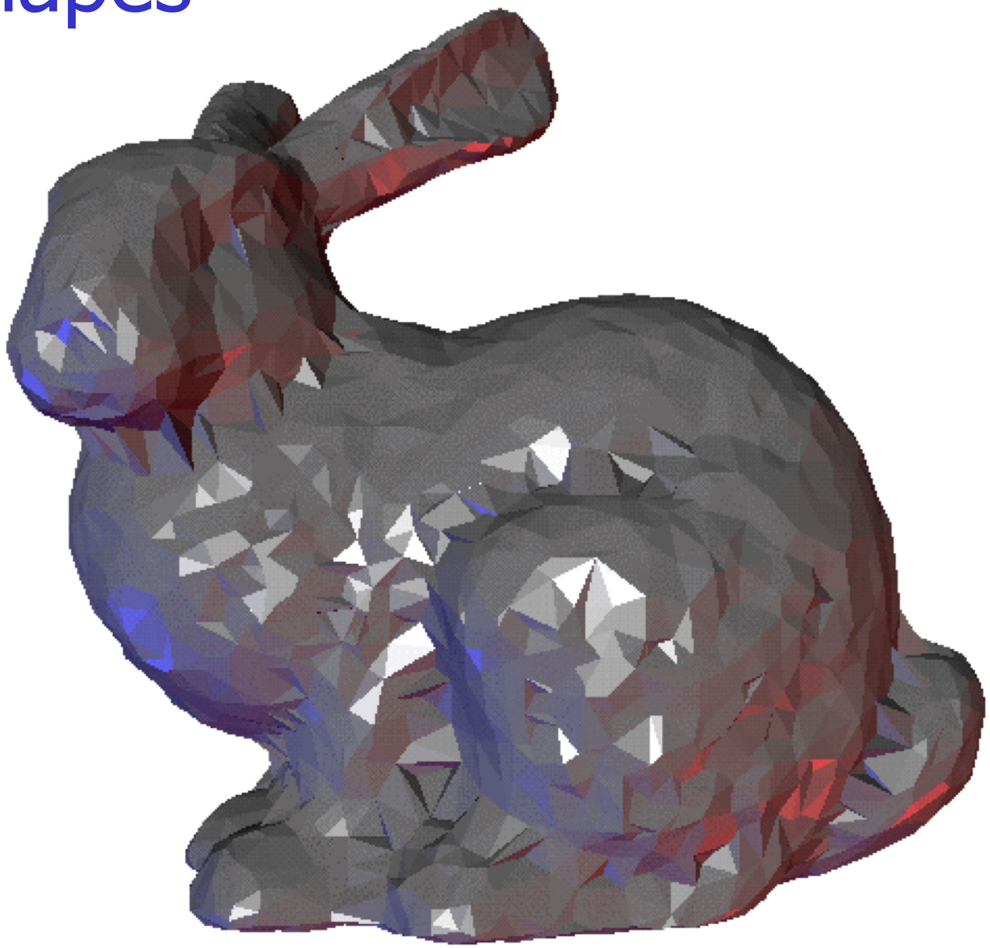
- $\alpha = \infty$: konvexe Hülle



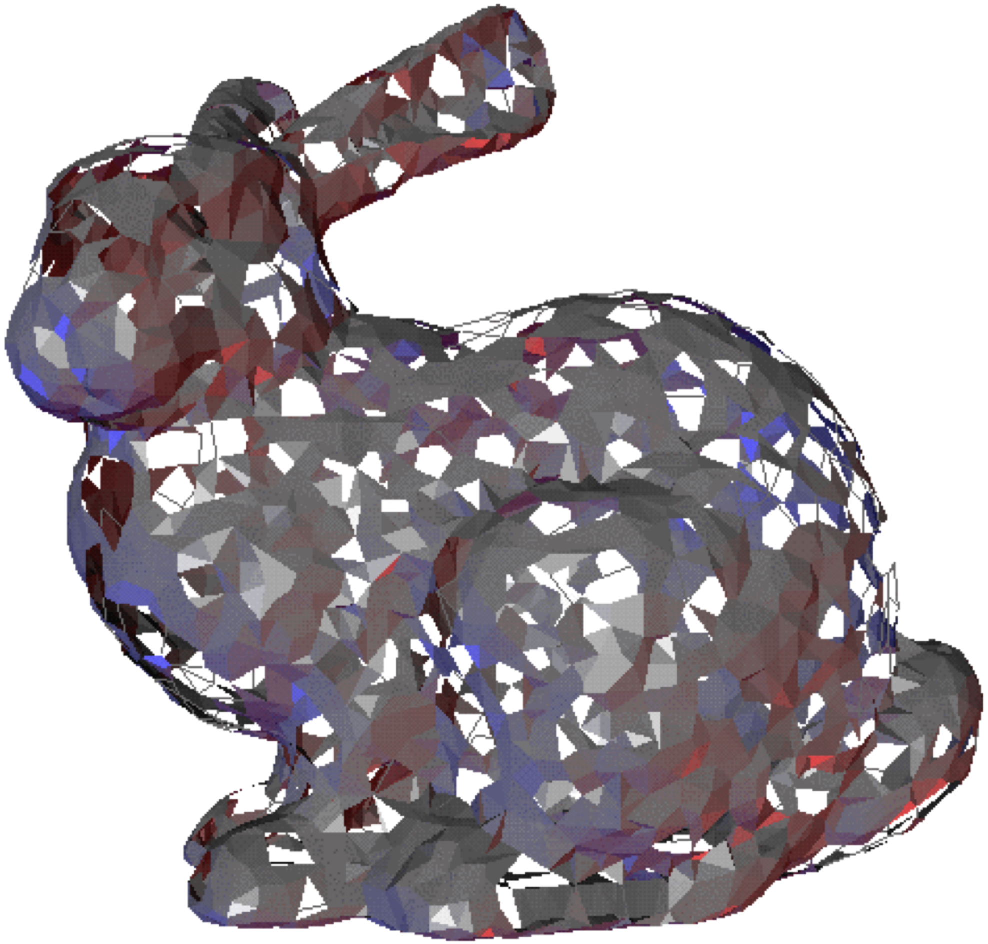
Alpha-shapes



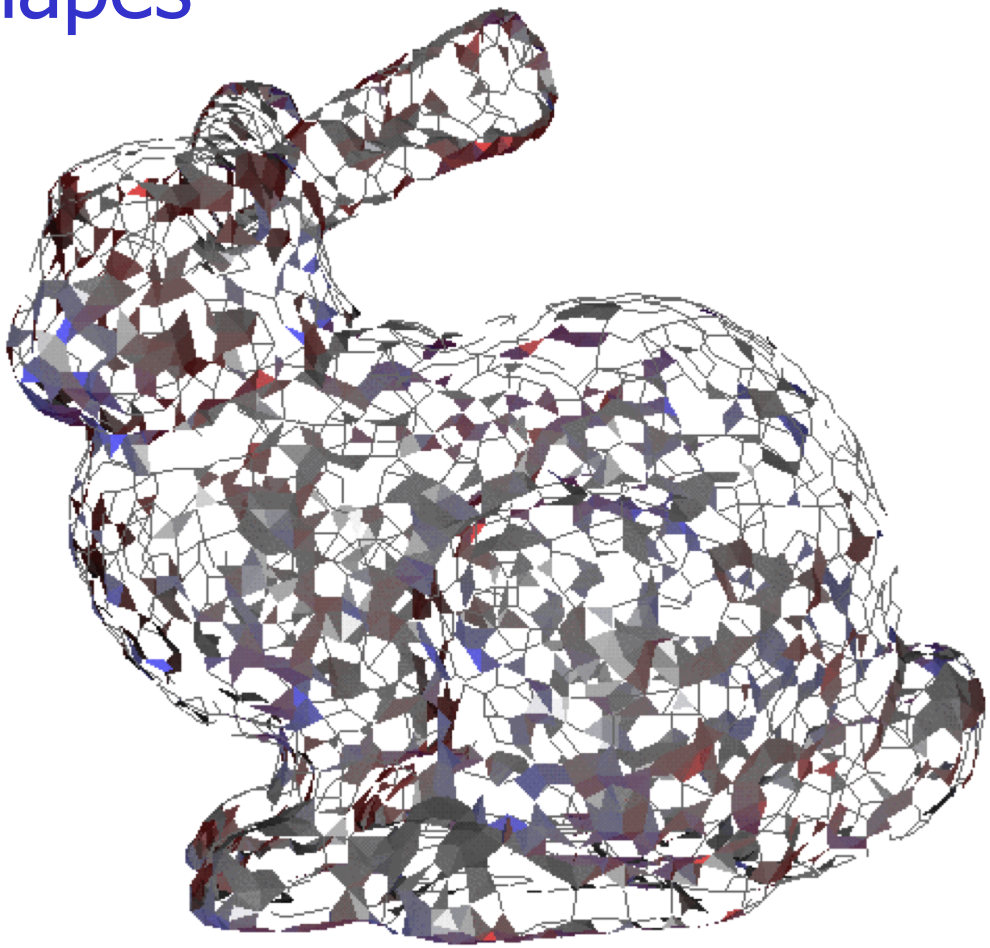
Alpha-shapes



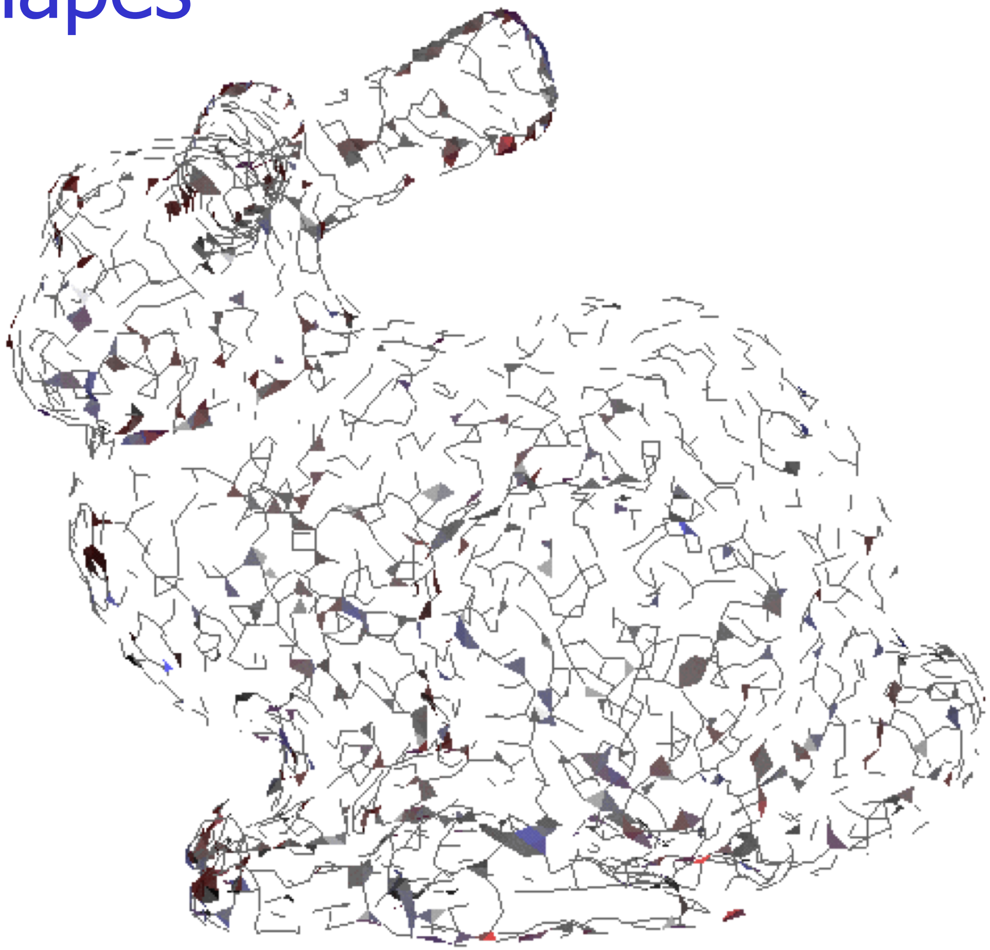
Alpha-shapes



Alpha-shapes



Alpha-shapes



$$\alpha = 0$$

Gibt originale Punktwolke