

Vorlesungen

„Einführung in die Datenvisualisierung“

**„Angewandte Datenvisualisierung für
Medizinphysiker“**

Kapitel E.

Visualisierung räumlicher Daten

Dr. Frank Weichert

Informatik VII

Technische Universität Dortmund

<http://ls7-www.cs.uni-dortmund.de>

Nur zum persönlichen Gebrauch durch Mitglieder der Universität Dortmund

E.1. Räumliche Daten

- E.1.1. Definition
- E.1.2. Repräsentationsformen
 - E.1.2.1. Punktbasierte Repräsentation
 - E.1.2.2. Voxelbasierte Repräsentation
 - E.1.2.3. Rechtecksgitter
 - E.1.2.4. Gekrümmte Gitter
 - E.1.2.5. Unstrukturierte (irreguläre) Gitter

E.2. Alpha-Shapes

- E.2.1. Einleitung
- E.2.2. Definitionen
- E.2.3. Alpha-Hülle
- E.2.4. Alpha-Form
- E.2.5. Randstreckung
- E.2.6. Bemerkungen
- E.2.7. Beispiel

E.3. Delaunay Triangulierung

- E.3.1. Zellkomplexe
 - E.3.1.1. Simplizes
 - E.3.1.2. Triangulierung
- E.3.2. Delaunay Triangulierung
 - E.3.2.1. Definitionen
 - E.3.2.2. Bemerkungen
 - E.3.2.3. Eigenschaften
 - E.3.2.4. Delaunay-Triangulierung/konvexe Hülle
 - E.3.2.5. d -dimensionale Delaunay-Triangulierung
 - E.3.2.6. Algorithmus „d-dimensionale konvexe Hülle“
 - E.3.2.7. Beneath-Beyond-Algorithmus
 - E.3.2.8. Bemerkungen

E.4. Netzreduktionsverfahren

- E.4.1. Elementanzahlreduktion
- E.4.2. Fehlermetriken

E.5. Volumenvisualisierung

- E.5.1. Einführung
- E.5.2. Segmentierung
- E.5.3. Marching-Cubes-Algorithmus
 - E.5.3.1. Grundalgorithmus
 - E.5.3.2. Elementare Konfigurationen
 - E.5.3.3. Berechnung des Unterteilungspunkts
 - E.5.3.4. Bemerkungen
 - E.5.3.5. Adaptives Marching-Tetrahedron-Verfahren
 - E.5.3.6. Auflösung von Mehrdeutigkeiten

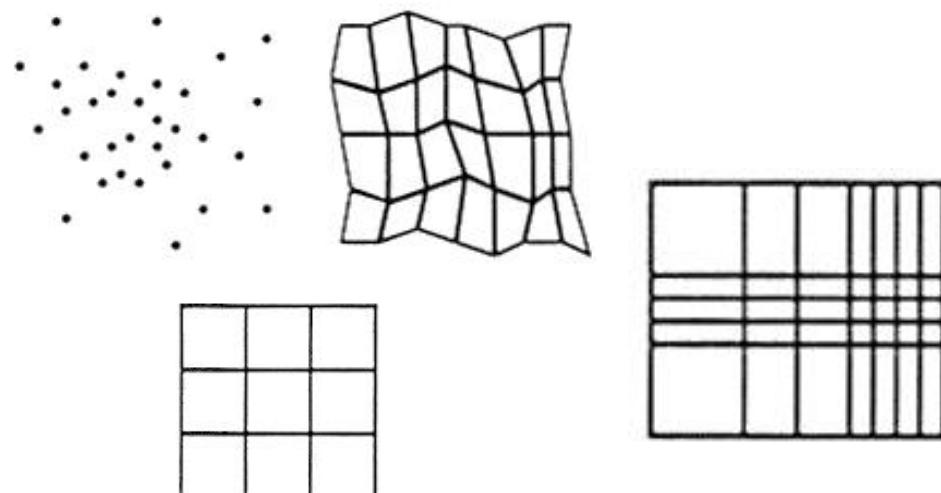
Datensatz

repräsentiert eine Funktion $f : G \rightarrow \mathbb{R}^m$, $G \subseteq \mathbb{R}^d$

wobei G ein d -dimensionales Gebiet im d -dimensionalen Raum ist.

Repräsentationsformen für $d = 2$ und $d=3$ (analog für $d > 3$):

- Punkte
- Voxel
- Rechtecksgitter
- gekrümmtes Gitter
- unstrukturiertes Gitter



Punktbasierte Repräsentation:

f wird durch Funktionswerte an einer endlichen Menge S von d -dimensionalen Punkten repräsentiert:

$$V = \{(x, y, z, f) \mid f = f(x, y, z), (x, y, z) \in S\}$$

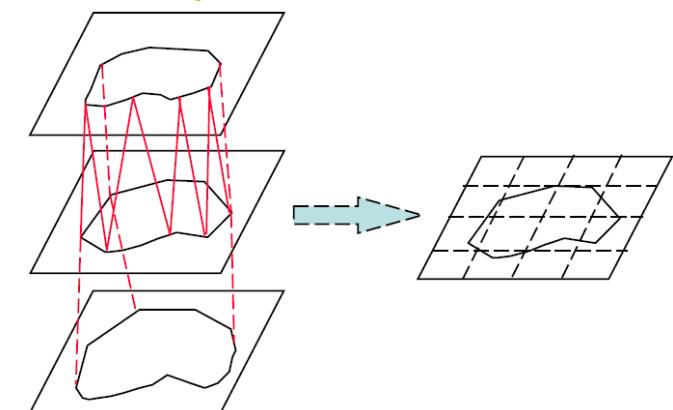


Visualisierung:

Interpolation mit Streudateninterpolationsverfahren und Visualisierung der resultierenden kontinuierlichen Funktion entsprechend derer Repräsentation, z.B. als voxelbasierte Repräsentation

Triangulierung über Oberflächenkonturen (bei 2D+t Daten):

- Extraktion von Konturen auf jeder Schicht
- Konstruktion eines Dreiecksnetzes zwischen zwei Schichten s_i und s_{i+1}

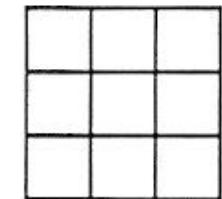


Voxelbasierte Repräsentation:

f wird durch Funktionswerte an den Eckpunkten eines regulären Gitters S repräsentiert:

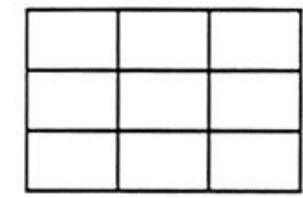
$$V = \{(i, j, k, f) \mid f = f(v), v = (x_i, y_j, z_k) \in S\}$$

$$S = \{(x_i, y_j, z_k) \mid (i, j, k) \in I_p \times I_q \times I_r\}$$



wobei

$$p, q, r \geq 0, p, q, r \in \mathbb{N}, I_n := \{0, 1, \dots, n\},$$



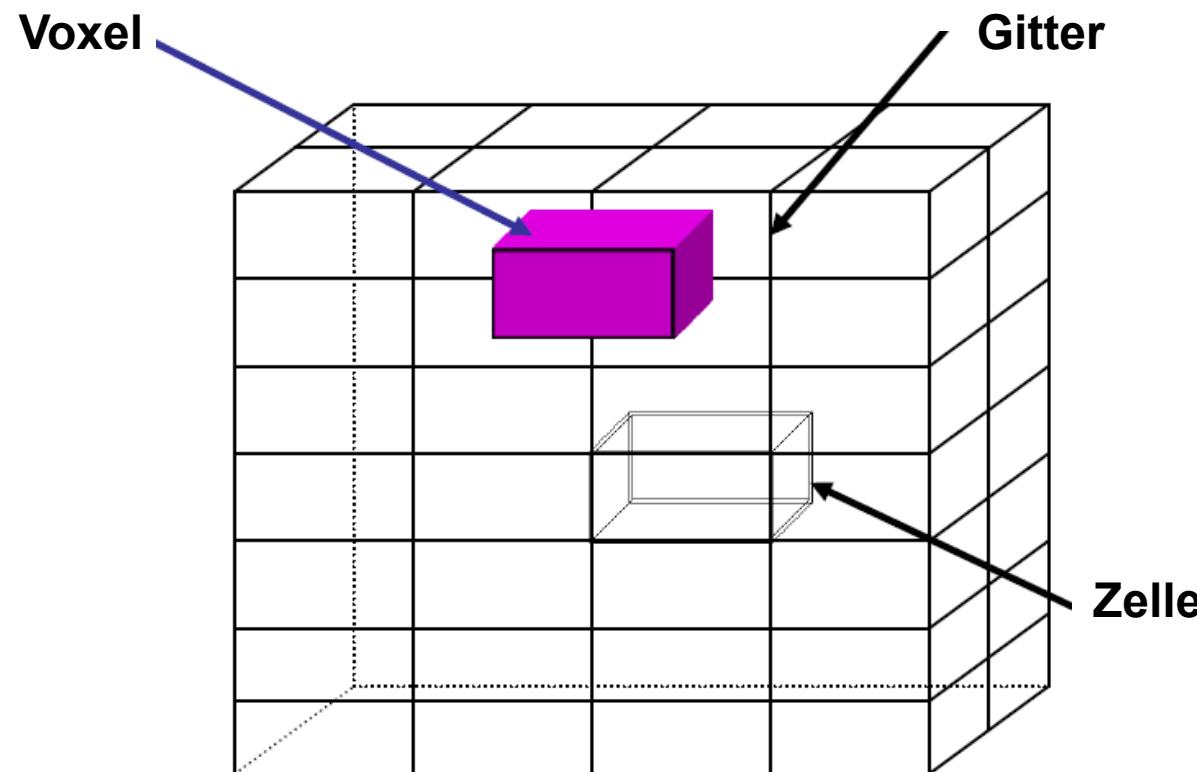
2d-Darstellung

$$x_{i+1} - x_i = \Delta x, y_{j+1} - y_j = \Delta y, z_{k+1} - z_k = \Delta z,$$

$$\Delta x, \Delta y, \Delta z > 0.$$

Voxelbasierte Repräsentation:

Die Zellen des Gitters werden als Voxel (Volumenelemente) bezeichnet.

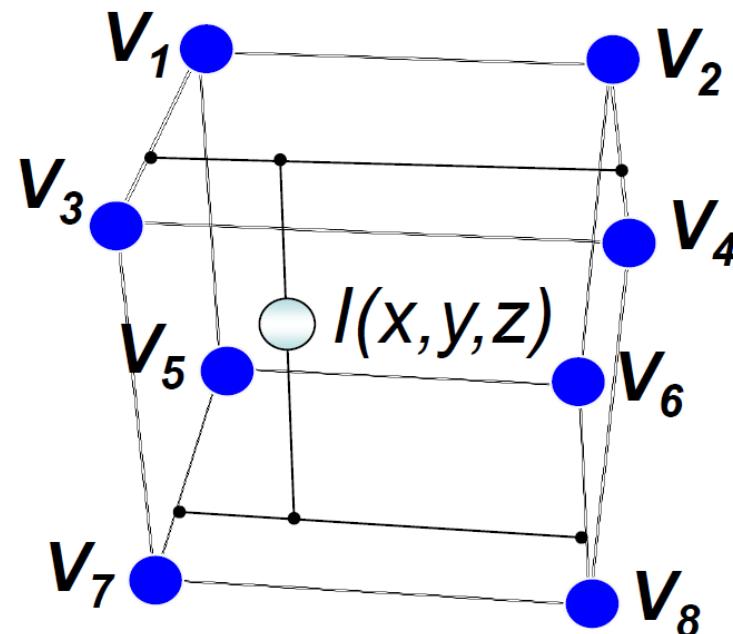


Voxel-Modelle:

- Binär
- Graustufen: Intensitäten, z.B. 12 Bits/Voxel
- Generalisiert: Attribute, z.B. Segmente, Farbe, Material
- Semantisch: Eigenschaften der zugehörigen Objekte und deren Beziehung zur Umgebung in einer semantische Repräsentationsform

Interpolation:

- Nächster Nachbar Interpolation
- Tri-lineare Interpolation
- Tri-kubische Interpolation



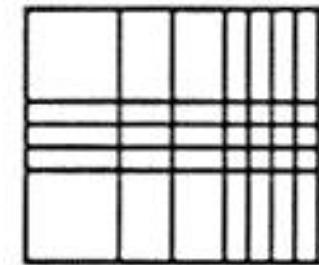
Repräsentation durch ein Rechtecksgitter:

f wird durch Funktionswerte an den Eckpunkten eines Gitters S repräsentiert, das aus rechtwinkligen Zellen besteht, die jedoch unterschiedliche Ausdehnung haben können:

$$V = \{(i, j, k, f) \mid f = f(v), v = (x_i, y_j, z_k) \in S\}$$

$$S = \{(x_i, y_j, z_k) \mid (i, j, k) \in I_p \times I_q \times I_r\}$$

wobei



2d-Darstellung

$$p, q, r \geq 0, p, q, r \in \mathbb{N}, I_n := \{0, 1, \dots, n\},$$

$$x_{i+1} - x_i = \Delta x_i, y_{j+1} - y_j = \Delta y_j, z_{k+1} - z_k = \Delta z_k,$$

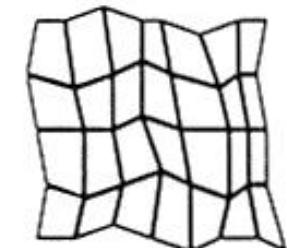
$$\Delta x_i, \Delta y_j, \Delta z_k > 0.$$

Repräsentation durch ein gekrümmtes Gitter:

\mathbf{f} wird durch Funktionswerte an den Eckpunkten eines Gitters S repräsentiert, das sich durch nichtlineare stetige Deformation aus einem Rechtecksgitter ergibt:

$$V = \{(i, j, k, f) \mid f = f(w), w = d(x_i, y_j, z_k) \in S\}$$

$$S = \{d(x_i, y_j, z_k) \mid (i, j, k) \in I_p \times I_q \times I_r\}$$



wobei $p, q, r \geq 0$, $p, q, r \in \mathbb{N}$, $I_n := \{0, 1, \dots, n\}$,

2d-Darstellung

$$x_{i+1} - x_i = \Delta x_i, y_{j+1} - y_j$$

$$= \Delta y_j, z_{k+1} - z_k = \Delta z_k, \Delta x_i, \Delta y_j, \Delta z_k > 0,$$

$d : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ eine stetige Deformationsfunktion.

Das durch (i, j, k) gegebene Gitter wird auch als *Berechnungsraum*, das deformierte Gitter als *physikalischer Raum* bezeichnet.

Repräsentation durch unstrukturierte (irreguläre) Gitter:

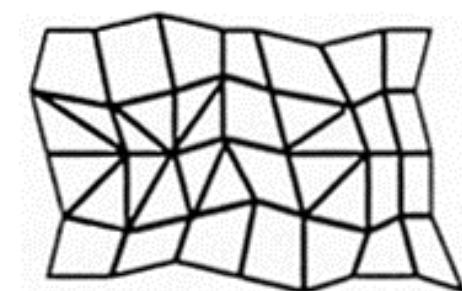
f wird durch Funktionswerte an den Knoten eines polygonalen oder simplizialen Komplexes repräsentiert.

Ein simplizialer Komplex setzt sich für $d = 3$ aus gegebenen abgeschlossenen Tetraedern, Dreiecken, Kanten und Knoten zusammen, die entweder disjunkt sind oder einen Durchschnitt haben, der von solchen gegebenen Elementen gebildet wird.

Für $d \neq 3$ sind Simplizes der Dimensionen von 0 bis d die als Elemente gegeben.

Ein polygonaler Komplex ist analog definiert, wobei anstelle von Simplizes Polytope verwendet werden.

Ein k -dimensionales Polytop ist eine solide Punktmenge, die sich als simplizialer Komplex darstellen lässt.



2d-Darstellung

E.1. Räumliche Daten

- E.1.1. Definition
- E.1.2. Repräsentationsformen
 - E.1.2.1. Punktbasierte Repräsentation
 - E.1.2.2. Voxelbasierte Repräsentation
 - E.1.2.3. Rechtecksgitter
 - E.1.2.4. Gekrümmte Gitter
 - E.1.2.5. Unstrukturierte (irreguläre) Gitter

E.2. Alpha-Shapes

- E.2.1. Einleitung
- E.2.2. Definitionen
- E.2.3. Alpha-Hülle
- E.2.4. Alpha-Form
- E.2.5. Randstreckung
- E.2.6. Bemerkungen
- E.2.7. Beispiel

E.3. Delaunay Triangulierung

- E.3.1. Zellkomplexe
 - E.3.1.1. Simplizes
 - E.3.1.2. Triangulierung
- E.3.2. Delaunay Triangulierung
 - E.3.2.1. Definitionen
 - E.3.2.2. Bemerkungen
 - E.3.2.3. Eigenschaften
 - E.3.2.4. Delaunay-Triangulierung/konvexe Hülle
 - E.3.2.5. d -dimensionale Delaunay-Triangulierung
 - E.3.2.6. Algorithmus „d-dimensionale konvexe Hülle“
 - E.3.2.7. Beneath-Beyond-Algorithmus
 - E.3.2.8. Bemerkungen

E.4. Netzreduktionsverfahren

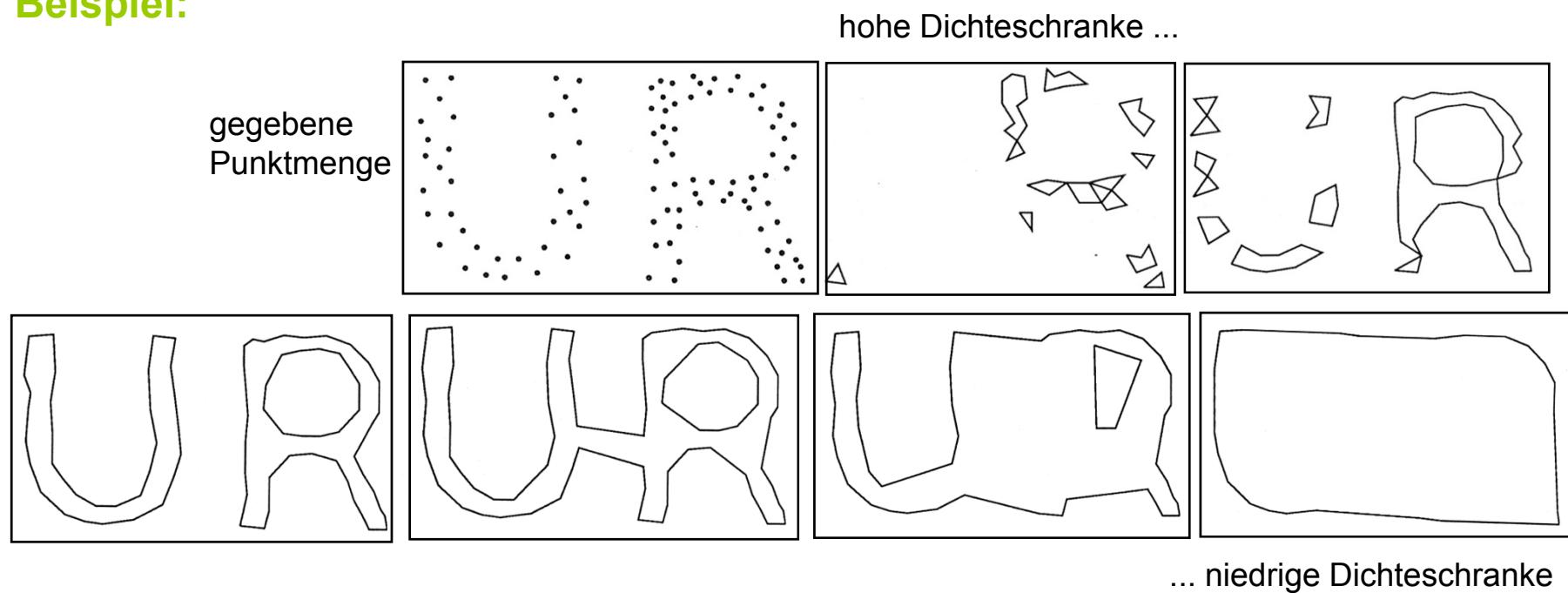
- E.4.1. Elementanzahlreduktion
- E.4.2. Fehlermetriken

E.5. Volumenvisualisierung

- E.5.1. Einführung
- E.5.2. Segmentierung
- E.5.3. Marching-Cubes-Algorithmus
 - E.5.3.1. Grundalgorithmus
 - E.5.3.2. Elementare Konfigurationen
 - E.5.3.3. Berechnung des Unterteilungspunkts
 - E.5.3.4. Bemerkungen
 - E.5.3.5. Adaptives Marching-Tetrahedron-Verfahren
 - E.5.3.6. Auflösung von Mehrdeutigkeiten

Gegenstand:

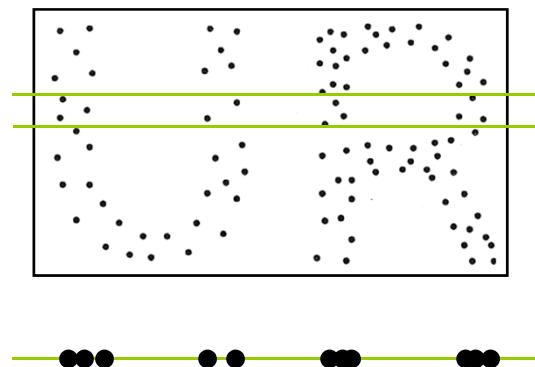
- Bestimmung des Randes von Gebieten mit einer Punktdichte größer als eine vorgegebene Schranke in einer Punktmenge
- Vorgabe der Schranke über einen Parameter α .

Beispiel:

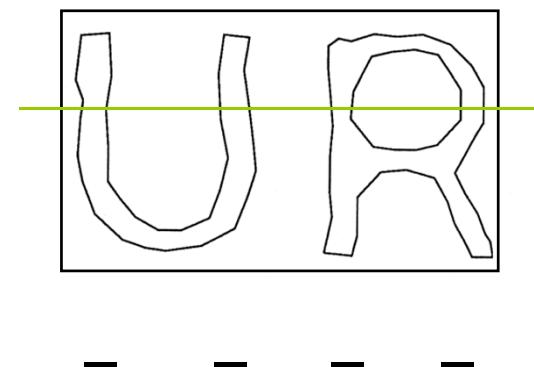
Anwendung bei der Visualisierung:

Visualisierung von Eigenschaften einer hochdimensionalen Punktemenge in Schnitten

Beispiel:



Hyperslice



Geometrischer Schnitt

***d*-Simplex:**

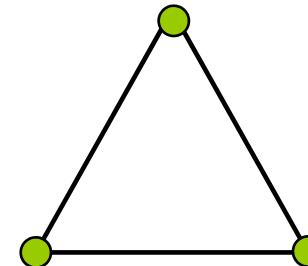
- Ein regulärer d -dimensionaler Simplex S_d ist ein Polytop im R^{d+1} , dessen konvexe Hülle den $d+1$ Einheitsvektoren entspricht.
- Simplex S_d hat $d+1$ Eckpunkte und $d+1$ Seitensimplexe der Dimension $d-1$.
- somit: 0-dimensionaler Simplex ist ein Punkt, ein 1-dimensionaler Simplex ist eine Linie, ein 2-dimensionaler Simplex ein gleichseitiges Dreieck, ein 3-dimensionaler Simplex ein Tetraeder.



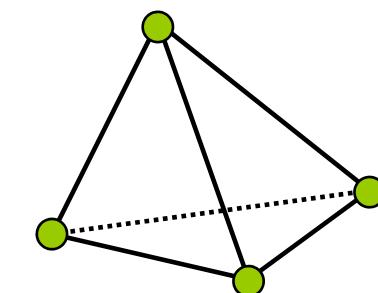
Punkt
0-Simplex



Linie
1-Simplex



Dreieck
2-Simplex



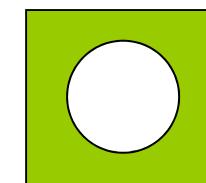
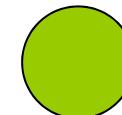
Tetraeder
3-Simplex

α -Hülle und α -Form

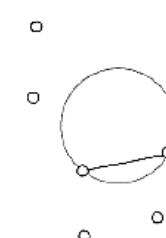
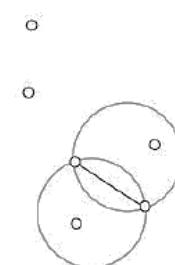
Gegeben: n Punkte $p_1, p_2, \dots, p_n \in \mathbb{R}^2$, $\alpha \in \mathbb{R}$.

 α -Scheibe (allgemein α -Ball) :

- abgeschlossene Kreisscheibe mit Radius $1/\alpha$, falls $\alpha > 0$
 - abgeschlossene Halbebene, falls $\alpha = 0$,
 - abgeschlossenes Komplement der Kreisscheibe mit Radius $-1/\alpha$, falls $\alpha < 0$.
- **volle α -Scheibe:** eine α -Scheibe, die alle Punkte p_i enthält.

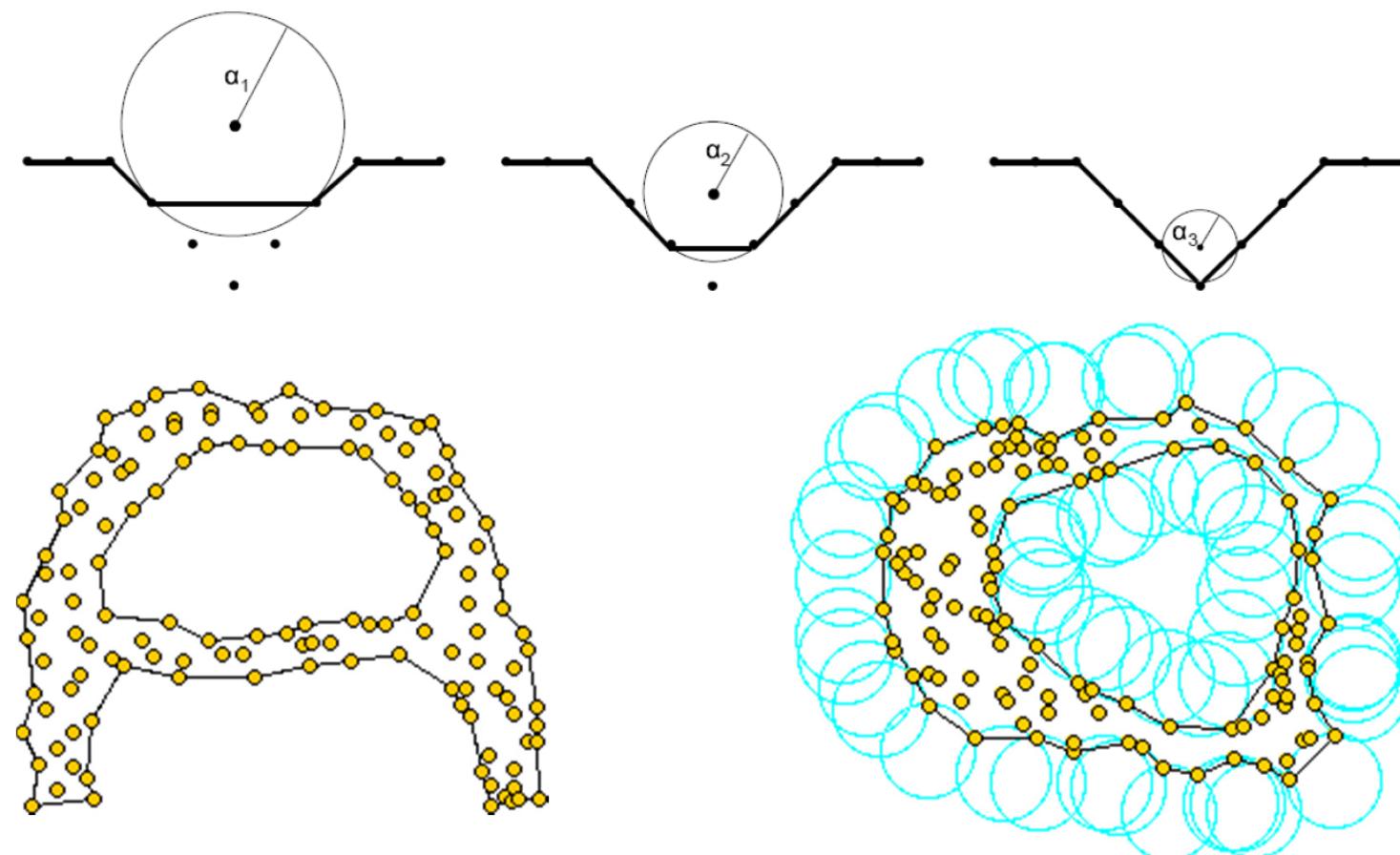


- **α -exponiert:** Ein k -Simplex ist α -exponiert, sofern eine zugehörige leere α -Scheibe existiert.

 *α -exponiert**nicht α -exponiert*

Intuitive Idee:

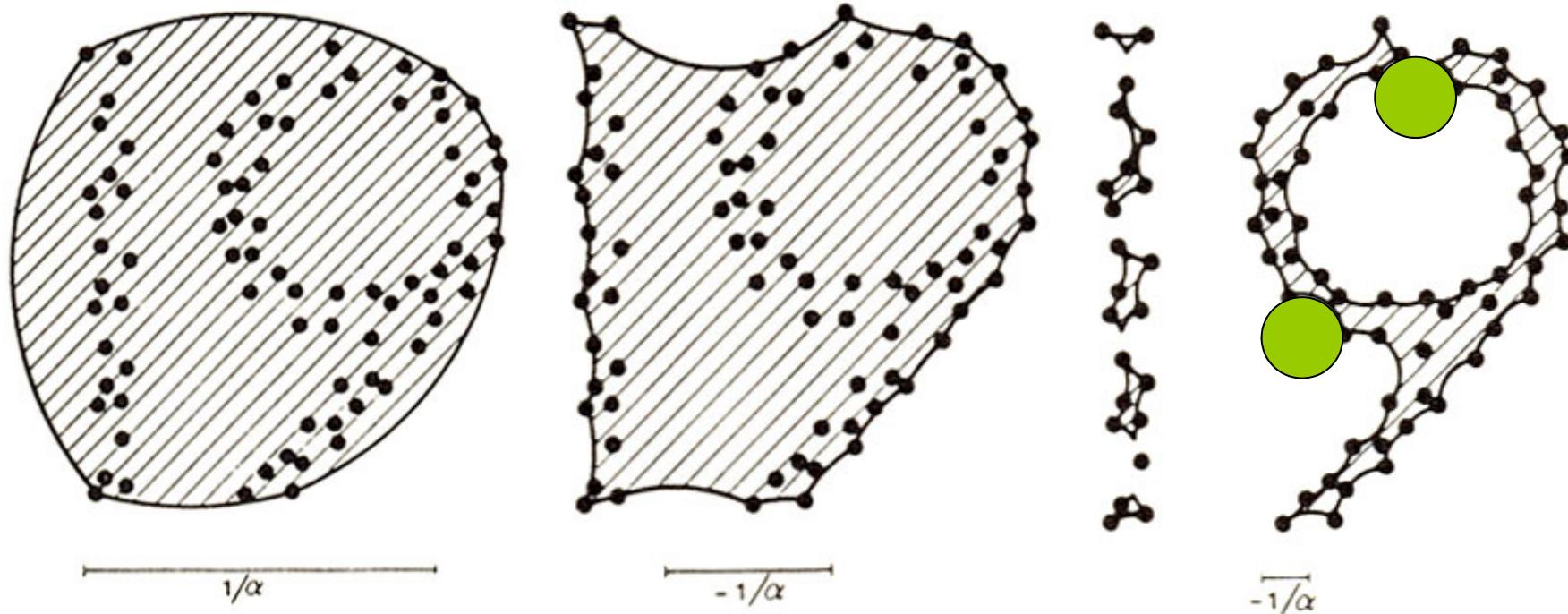
Eine Kreisschreibe (Kugel) b_α mit Radius α rollt über eine Punktwolke und beschreibt dabei Paare von Punkten.



α -Hülle:

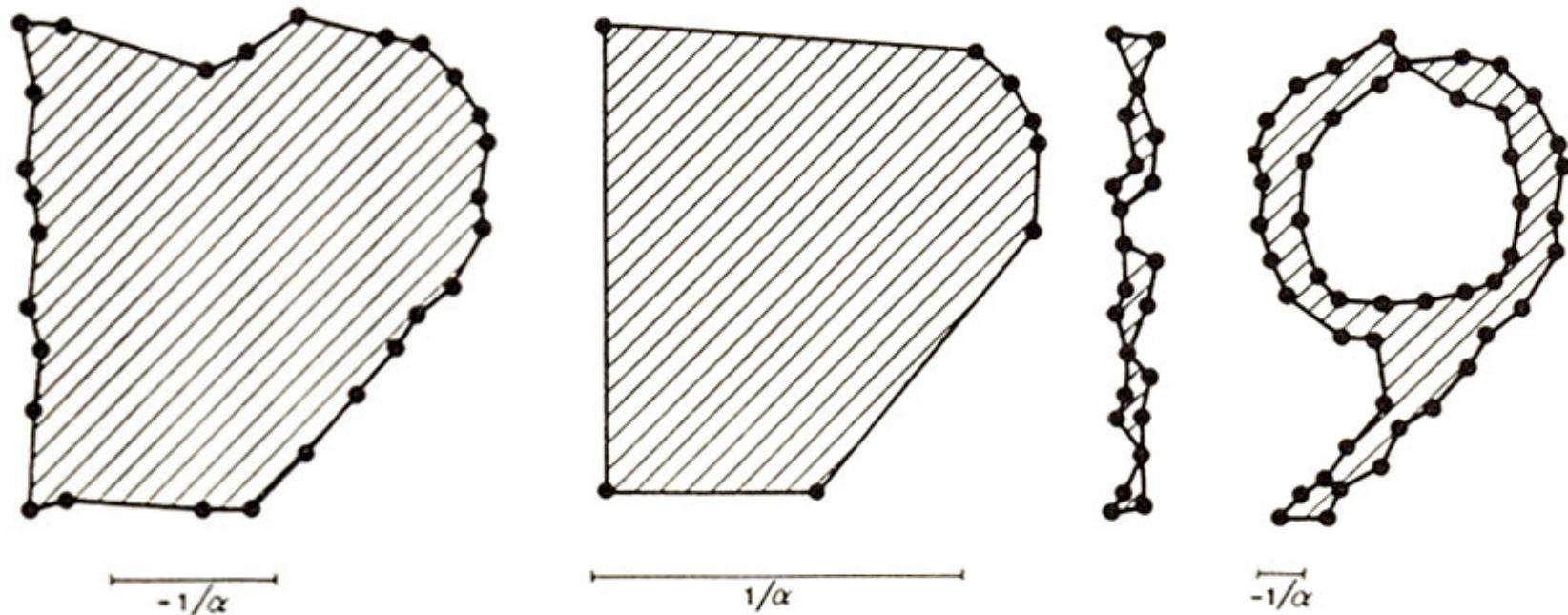
Der Durchschnitt aller vollen α -Scheiben der gegebenen Punkte.

Beispiel:



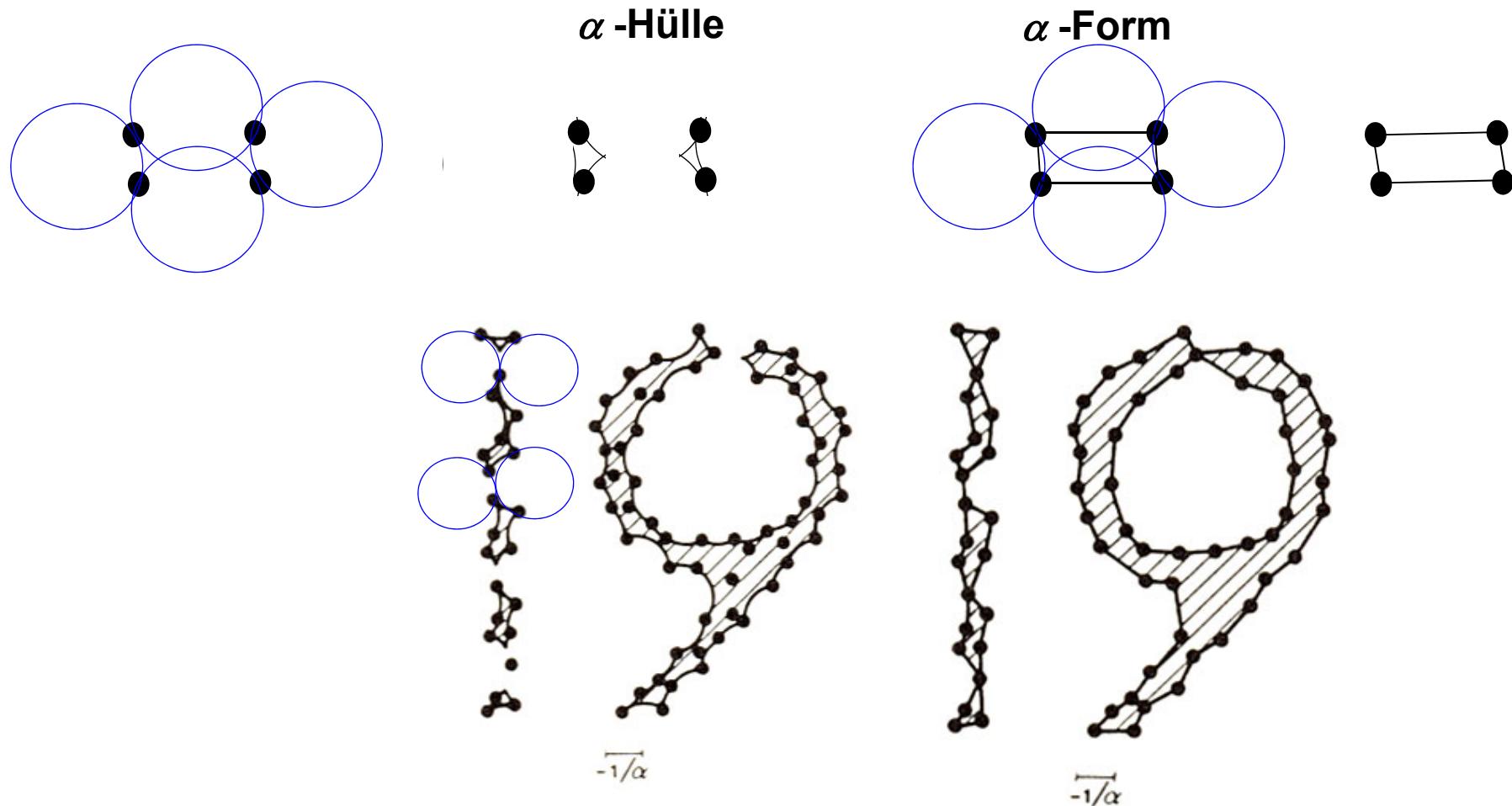
α -Form:

Der „gestreckte“ Rand aller volldimensionalen Zusammenhangskomponenten der α -Hülle.

Beispiele:

Streckung des Randes:

Einfügen einer Sehne zwischen den Kontaktpunkten eines Randkreises



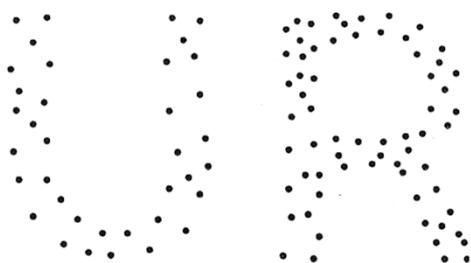
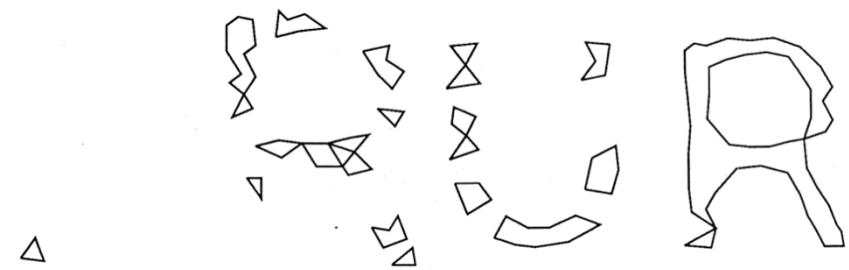
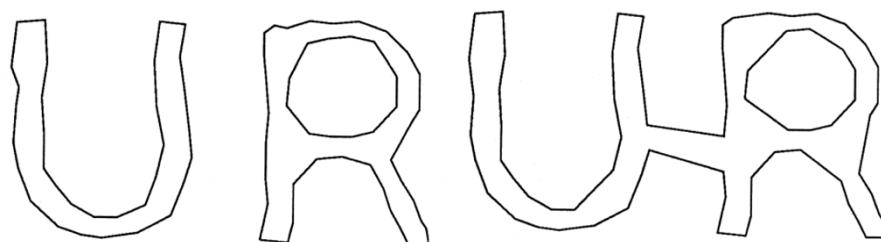
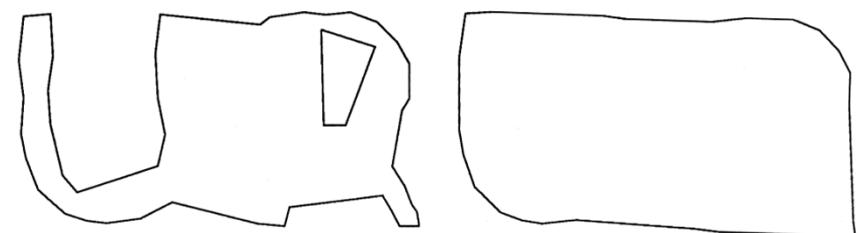
Bemerkungen:

- Die α -Hüllen und α -Formen stellen eine kombinatorische Interpolation dar, die durch die Dichte der gegebenen Punkte gesteuert wird. Eher dicht liegende Punkte fallen in eine gemeinsame Zusammenhangskomponente.
- Für $\alpha < 0$ wird die α -Form durch alle Kanten der Delaunay-Triangulierung induziert, die kürzer als $-2/\alpha$ sind und für die eines der inzidenten Dreiecke einen Umkreisradius $< -1/\alpha$ hat und das andere, falls existent, einen Radius $> -1/\alpha$ besitzt. Damit kann die α -Form in diesem Fall über die Delaunay-Triangulierung berechnet werden.
- Für $\alpha = 0$ wird die α -Form von den äußeren Kanten definiert, d.h. den Kanten der konvexen Hülle.
- Für $\alpha > 0$ kann die α -Form als eine Ausdünnung der konvexen Hülle verstanden werden.

Beispiel:

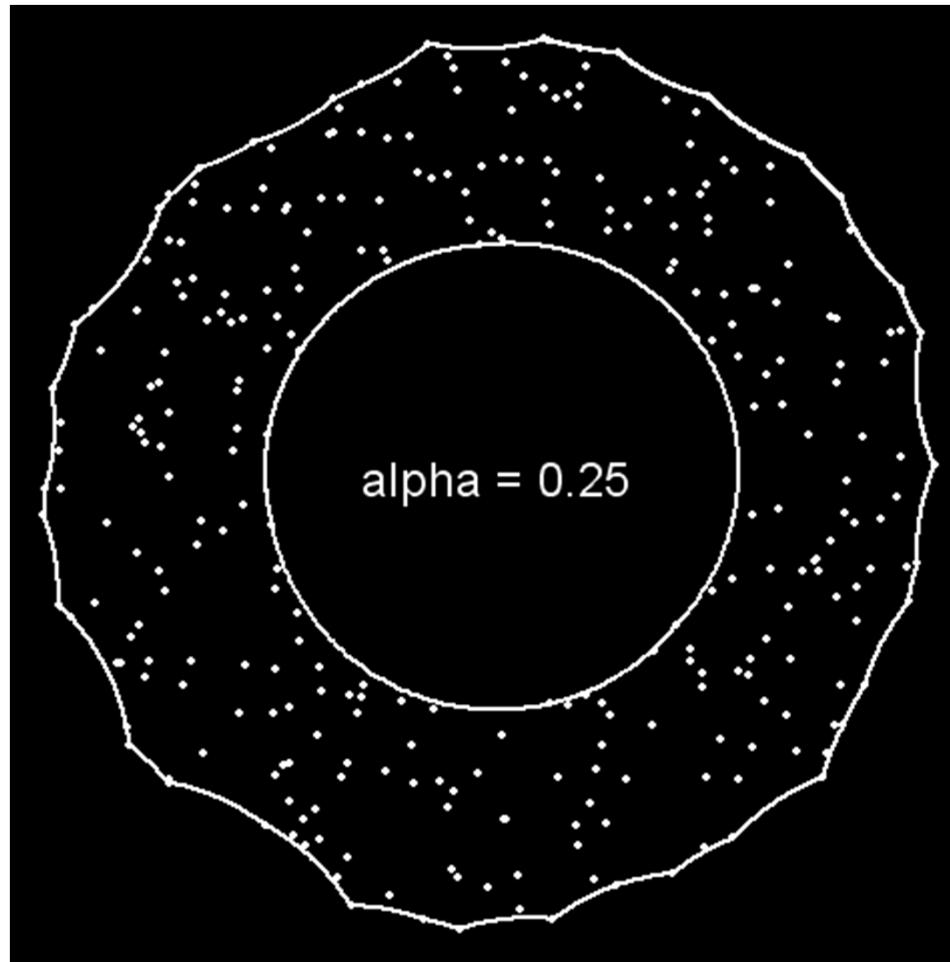
Evolution von α -Formen für monoton wachsende Werte von $-1/\alpha$.

Punktmenge

 α -Formen für $-1/\alpha = 5, 7.5$  α -Formen für $-1/\alpha = 10, 16$. α -Formen für $-1/\alpha = 20, 100$.

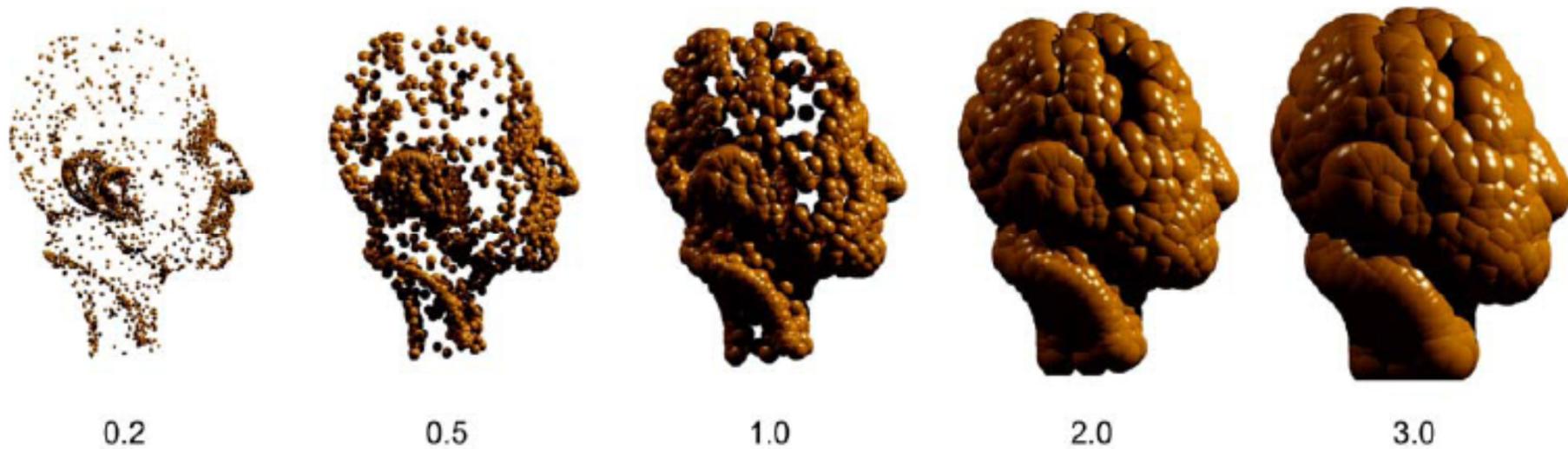
Beispiel:

Evolution von α -Formen für eine Folge von α -Werten.



Beispiel:

Evolution von α -Formen im 3D-Fall.



E.1. Räumliche Daten

- E.1.1. Definition
- E.1.2. Repräsentationsformen
 - E.1.2.1. Punktbasierte Repräsentation
 - E.1.2.2. Voxelbasierte Repräsentation
 - E.1.2.3. Rechtecksgitter
 - E.1.2.4. Gekrümmte Gitter
 - E.1.2.5. Unstrukturierte (irreguläre) Gitter

E.2. Alpha-Shapes

- E.2.1. Einleitung
- E.2.2. Definitionen
- E.2.3. Alpha-Hülle
- E.2.4. Alpha-Form
- E.2.5. Randstreckung
- E.2.6. Bemerkungen
- E.2.7. Beispiel

E.3. Delaunay Triangulierung

- E.3.1. Zellkomplexe
 - E.3.1.1. Simplizes
 - E.3.1.2. Triangulierung
- E.3.2. Delaunay Triangulierung
 - E.3.2.1. Definitionen
 - E.3.2.2. Bemerkungen
 - E.3.2.3. Eigenschaften
 - E.3.2.4. Delaunay-Triangulierung/konvexe Hülle
 - E.3.2.5. d -dimensionale Delaunay-Triangulierung
 - E.3.2.6. Algorithmus „d-dimensionale konvexe Hülle“
 - E.3.2.7. Beneath-Beyond-Algorithmus
 - E.3.2.8. Bemerkungen

E.4. Netzreduktionsverfahren

- E.4.1. Elementanzahlreduktion
- E.4.2. Fehlermetriken

E.5. Volumenvisualisierung

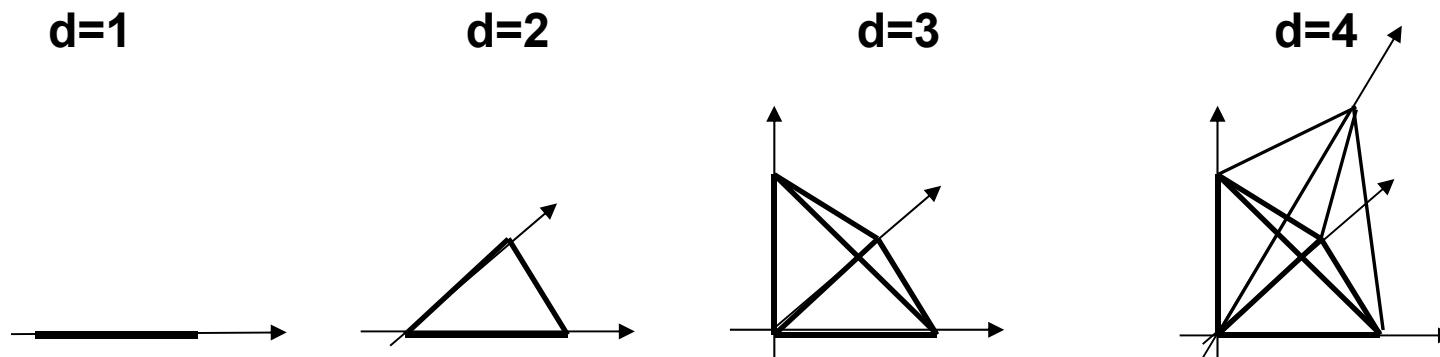
- E.5.1. Einführung
- E.5.2. Segmentierung
- E.5.3. Marching-Cubes-Algorithmus
 - E.5.3.1. Grundalgorithmus
 - E.5.3.2. Elementare Konfigurationen
 - E.5.3.3. Berechnung des Unterteilungspunkts
 - E.5.3.4. Bemerkungen
 - E.5.3.5. Adaptives Marching-Tetrahedron-Verfahren
 - E.5.3.6. Auflösung von Mehrdeutigkeiten

d-dimensionales Simplex:

Konvexe Hülle von $d+1$ Punkten im \mathbb{R}^d

Beispiel: Einheits-Simplizes

$$S_d := \{ \mathbf{p} : \mathbf{p} = (p_1, \dots, p_d), 0 \leq p_1, \dots, p_d \leq 1, p_1 + \dots + p_d \leq 1 \}$$

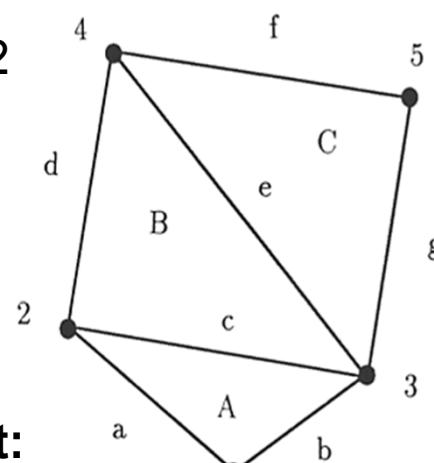


d-dimensionaler simplizialer Komplex

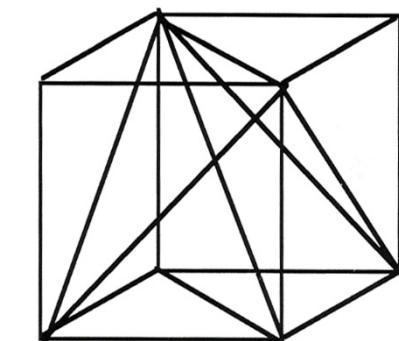
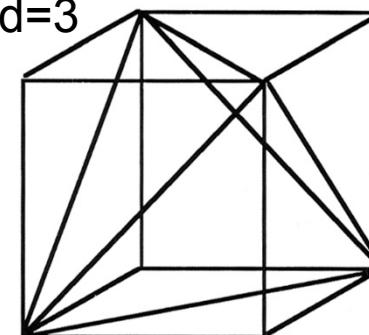
eine endliche Menge von d -dimensionalen Simplexen, so dass

- der Durchschnitt von je zwei Simplexen entweder leer oder ein Seiten-simplex beider Simplexe ist,
- mit jedem Simplex auch alle seine Seitensimplexe zur gegebenen Simplexmenge gehören.

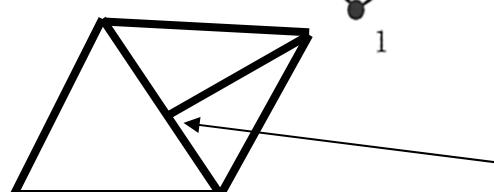
Beispiel: $d=2$



$d=3$



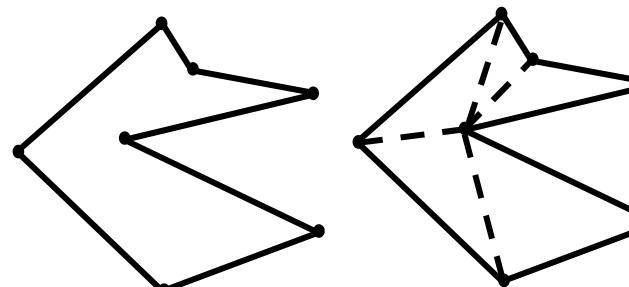
nicht korrekt:



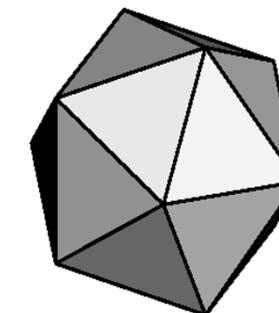
T-Stelle / „hängender Knoten“

Polyeder:

eine Punktmenge im \mathbb{R}^d , die durch eine d -dimensionale simpliziale Zerlegung beschrieben werden kann, dargestellt durch eine Hierarchie von Seitenpolyedern. Dabei setzt sich der Rand aus endlich vielen Seitenpolyedern zusammen, deren Rand wieder aus Seitenpolyedern und so fort, mit Eckpunkten als 0-dimensionale Polyeder.

Beispiele:

$d=2$

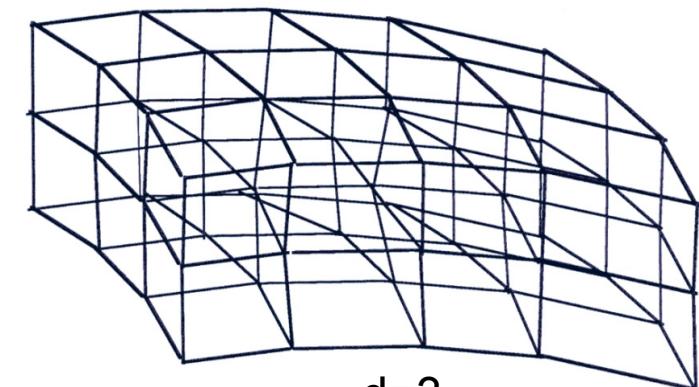
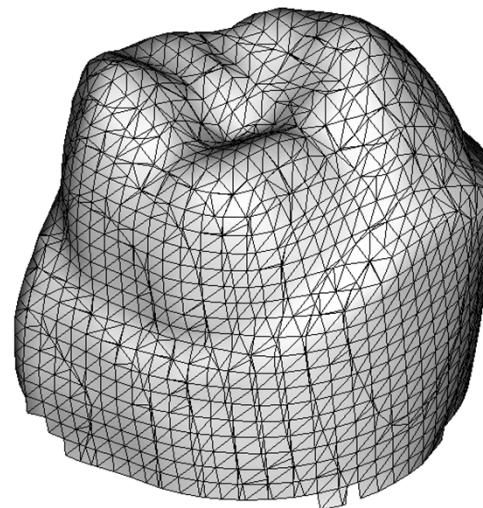
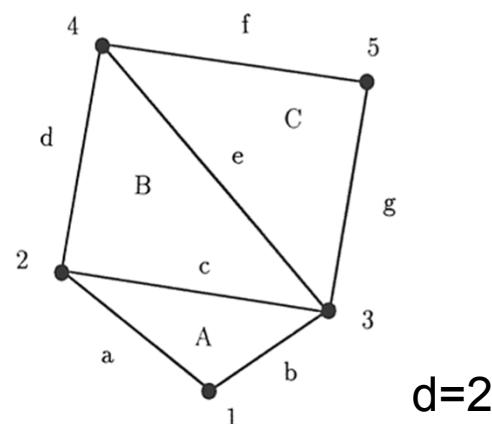


$d=3$

polyedrischer Komplex:

eine endliche Menge von Polyedern im mit den Eigenschaften, dass

- der Durchschnitt von je zwei Polyedern entweder leer oder ein Seitenpolyeder beider Polyeder ist,
- mit jedem Polyeder auch alle seine Seitenpolyeder zur gegebenen Polyedermenge gehören.

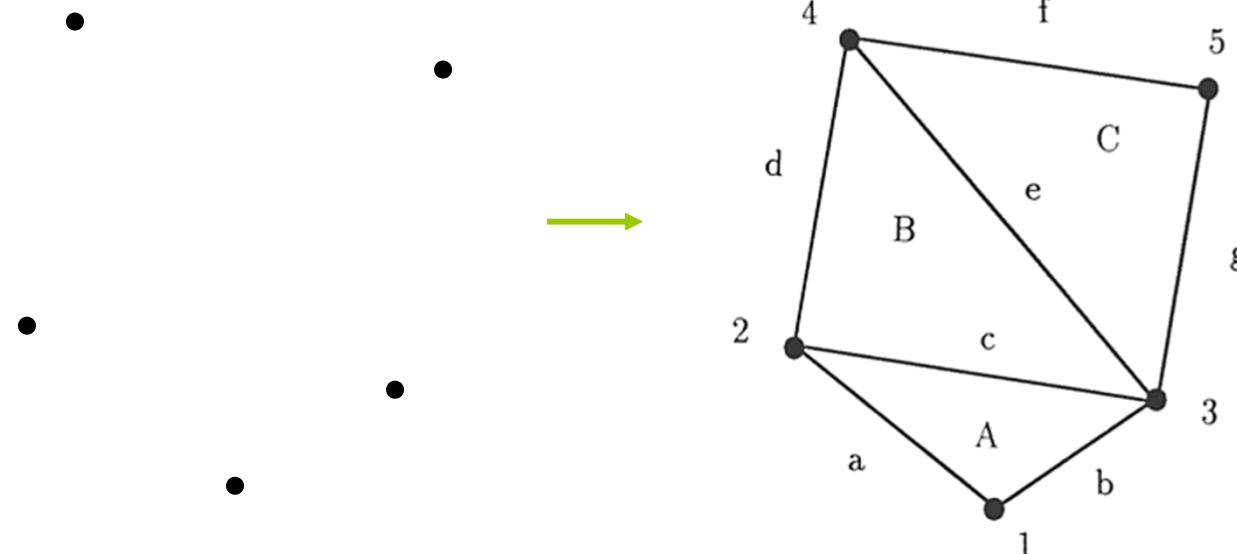
Beispiele:

Simplex

Ein Simplex $S \in \mathbb{R}^d$ ist die konvexe Hülle von $d+1$ Punkten p_i , die nicht alle in einer Hyperebene liegen.

Triangulierung einer Punktmenge $\{p_1, \dots, p_n \in \mathbb{R}^d\}$:

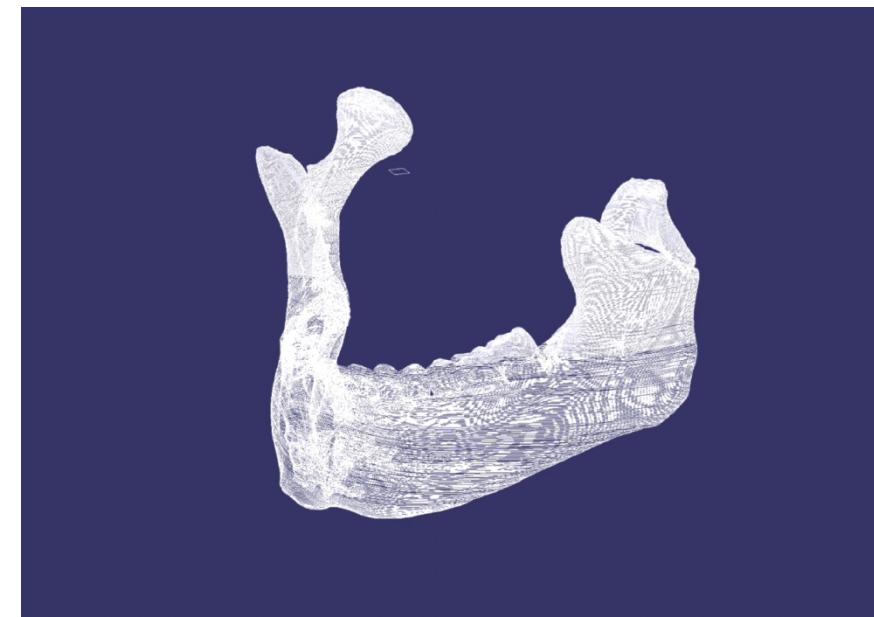
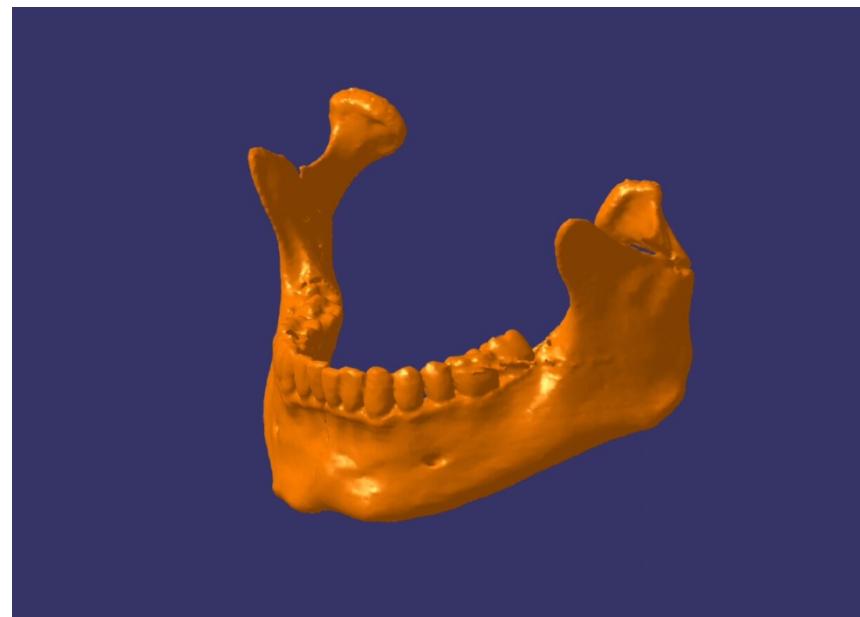
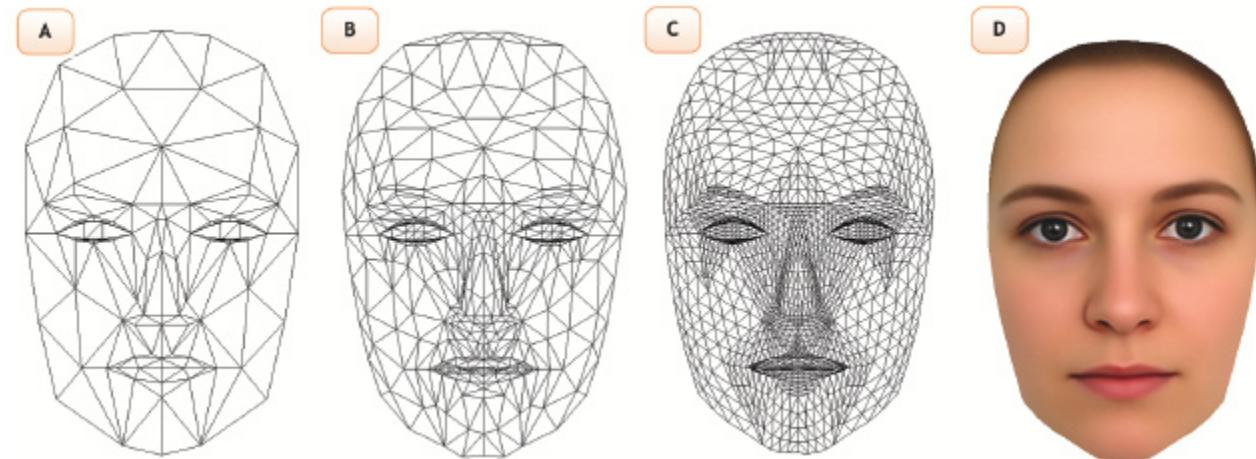
simpliziale Zerlegung der konvexen Hülle von p_1, \dots, p_n , die genau die gegebenen Punkte als Eckpunkte (nulldimensionale Simplexe) hat.

Beispiel:

E.3.1. Zellkomplexe

E.3.1.2. Triangulierung

Beispiel:

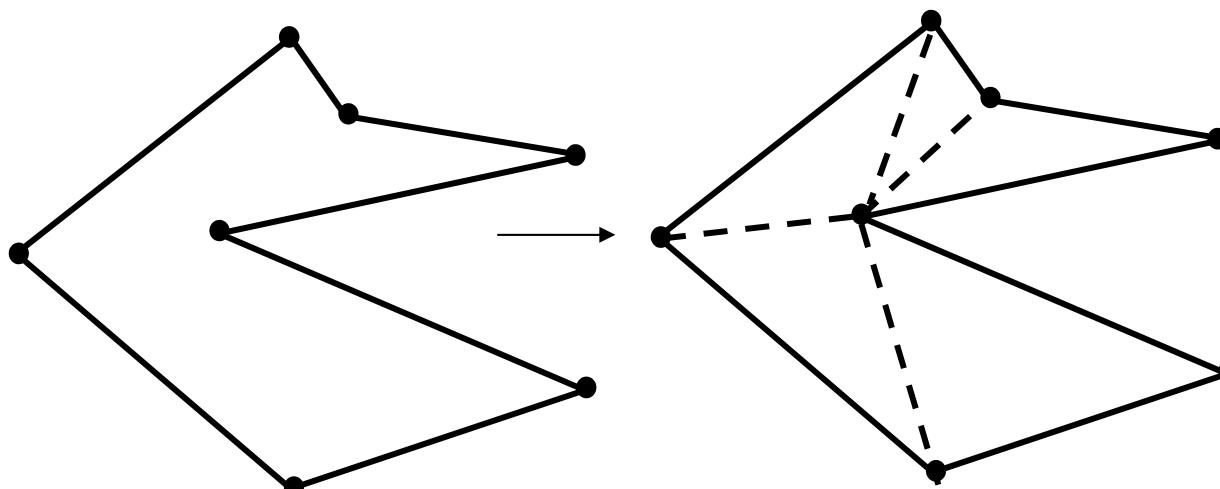


Polygontriangulierung

Gegeben: ein Polygon P .

Gesucht: eine Verbindung der Knoten des Polygons durch sich nicht schneidende Sehnen, so dass eine Zerlegung des Polygoninneren in Dreiecke induziert wird.

Beispiel



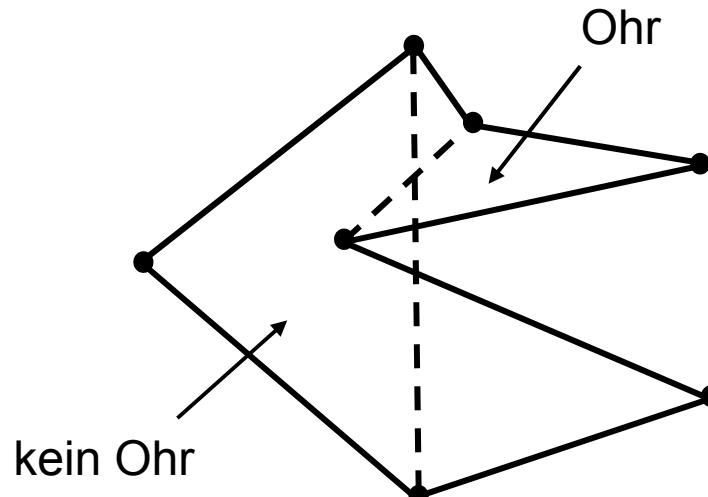
Triangulierung von Polygonen durch Ohrenabschneiden

Vorgehensweise:

sukzessives Entfernen von *Ohren* des gegebenen Polygons

Ohr: ein von einer Sehne induziertes Dreieck, das ganz im Inneren des Polygons liegt.

Behauptung (ohne Bew.): Jedes Polygon besitzt mindestens ein Ohr.



Beobachtung:

Ein von drei aufeinanderfolgenden Eckpunkten des Polygons gebildetes Dreieck ist schon ein Ohr, wenn es keinen nichtkonvexen Polygoneckpunkt enthält.

Delaunay-Triangulierung:

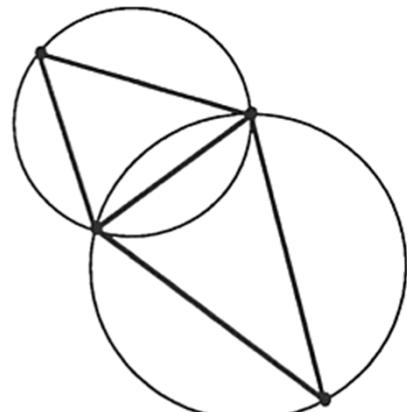
Eine Triangulierung, die nur aus Delaunay-Simplexen bezüglich der gegebenen Punkte besteht.

Delaunay-Simplex:

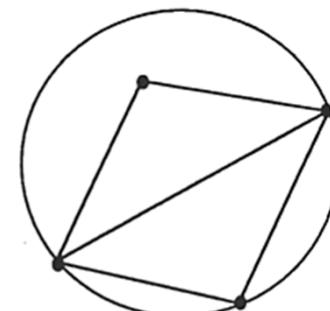
Ein Simplex mit k von n Punkten als Eckpunkte, $1 \leq k \leq \min\{n, d + 1\}$, der bezüglich der gegebenen Punkte die Umkugelbedingung erfüllt.

Beispiele:

Mögliche Triangulierungen von vier Punkten

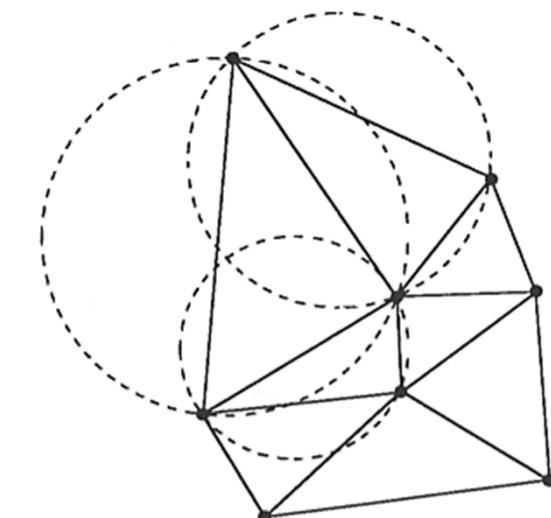


Delaunay-Triangulierung



Delaunay-Triangulierung

Delaunay-Triangulierung
von mehreren Punkten



Bemerkungen:

1. Im Zweidimensionalen sind Delaunay-Triangulierungen diejenigen, bei denen der kleinste Winkel über alle Dreiecke maximiert wird.
2. Delaunay-Triangulierungen in beliebigen Dimensionen können durch eine Berechnung der konvexen Hülle einer um eine Dimension höheren Punktmenge berechnet werden.

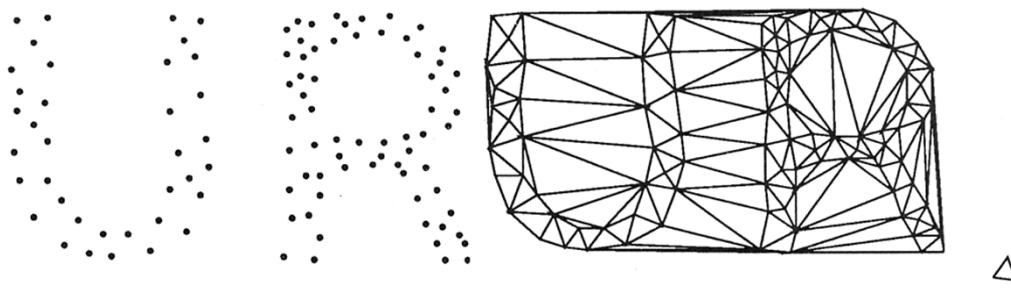
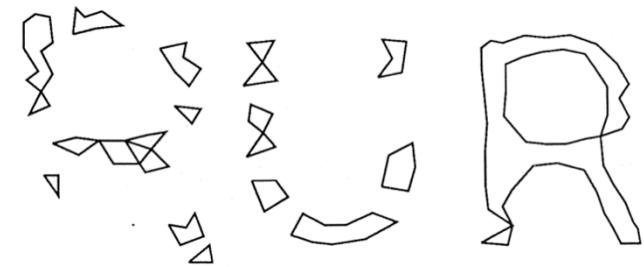
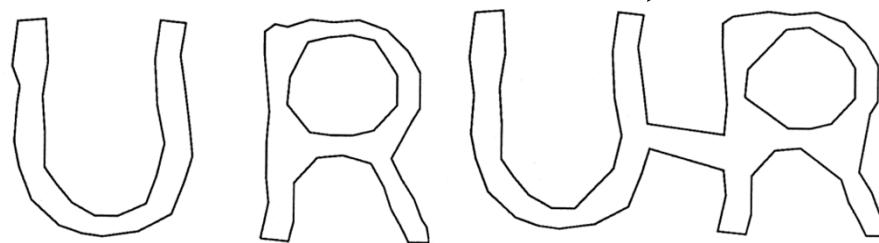
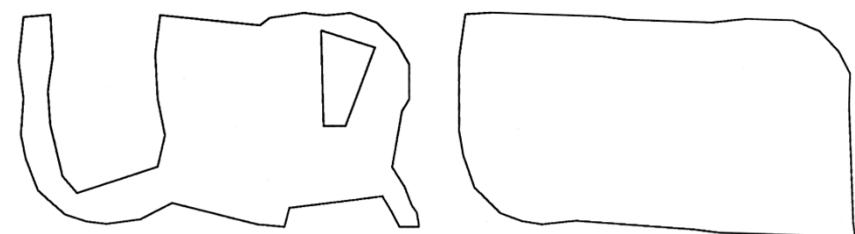
Literatur:

- F.P. Preparata, M.L. Shamos, Computational Geometry - An Introduction, Springer-Verlag, 1985
- T. Ottmann, P. Widmayer, Algorithmen und Datenstrukturen, B.I.-Wissenschaftsverlag, 1990

Bemerkung:

Für $\alpha < 0$ wird die α -Form durch alle Kanten der Delaunay-Triangulierung induziert, die kürzer als $-2/\alpha$ sind und für die eines der inzidenten Dreiecke einen Umkreisradius $< -1/\alpha$ hat und das andere, falls existent, einen Radius $> -1/\alpha$ besitzt.

Punktmenge und deren Delaunay-Triangulierung

 α -Formen für $-1/\alpha = 5, 7.5$  α -Formen für $-1/\alpha = 10, 16$  α -Formen für $-1/\alpha = 20, 100$ 

Eigenschaften von Delaunay-Triangulierungen

1. Existenz:

Zu jeder endlichen Punktmenge im \mathbb{R}^d gibt es eine Delaunay-Triangulierung.

2. Eindeutigkeit:

Sind keine $d + 2$ der gegebenen Punkte kosphärisch, dann ist die Delaunay-Triangulierung eindeutig.

3. Delaunay-Simplizes:

Sind keine $d + 2$ der gegebenen Punkte kosphärisch, dann

- ist eine Triangulierung genau dann eine Delaunay-Triangulierung, wenn jeder Simplex Delaunay-Simplex ist.
- sind die Simplizes der Delaunay-Triangulierung genau die Delaunay-Simplizes der gegebenen Punktmenge.

Beweis: s. nächste Folien

Transfomation Delaunay-Triangulierung/konvexe Hülle

Sei $T : \mathbb{R}^d \rightarrow \mathbb{R}^{d+1}$ definiert durch

$$T(\mathbf{p}) = \begin{pmatrix} \mathbf{p} \\ \sum_{i=1}^d p_i^2 \end{pmatrix}, \quad \mathbf{p} = (p_1, \dots, p_d).$$

Dann ist die Delaunay-Triangulierung zu den Punkten $\mathbf{p}_1, \dots, \mathbf{p}_n \in \mathbb{R}^d$ kombinatorisch äquivalent zu der Zellzerlegung, die von den d -dimensionalen Zellen der konvexen Hülle von $T(\mathbf{p}_1), \dots, T(\mathbf{p}_n)$ induziert werden, deren äußerer Normalenvektor eine negative $(d + 1)$ -Komponente hat.

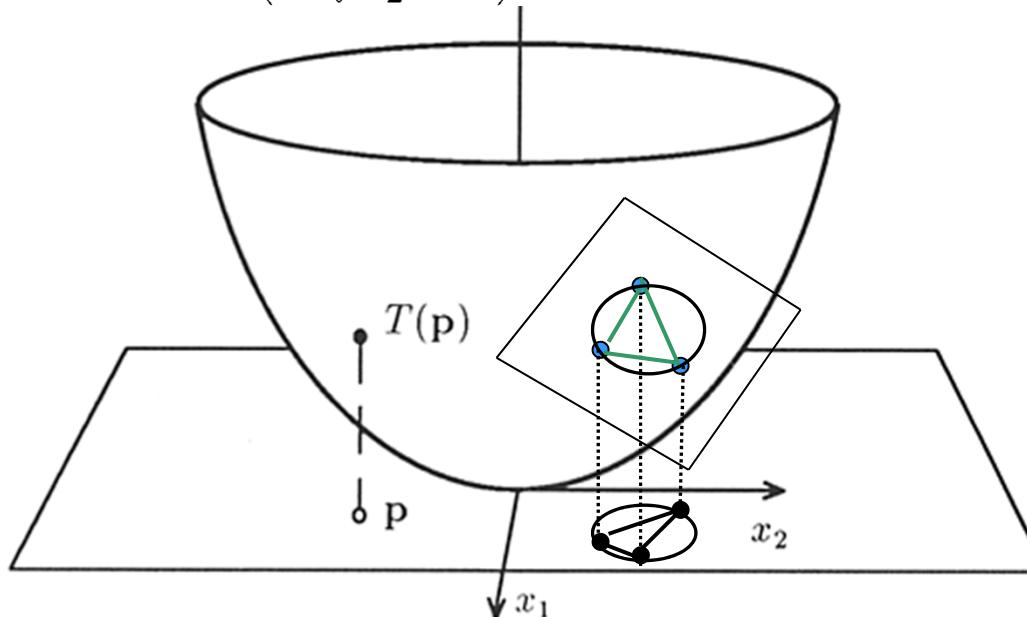
Die Delaunay-Triangulierung wird also von allen Zellen und deren Nachbarzellen der konvexen Hülle gebildet, welche von der Hyperebene aus sichtbar sind, die von den ersten d Koordinatenachsen aufgespannt wird.

Literatur:

H. Edelsbrunner, Algorithms in Combinatorial Geometry, Springer-Verlag, 1987

Beweisidee:

$$T(p) = \left(\frac{p}{\sum_{i=1}^d p_i^2} \right), \quad p = (p_1, \dots, p_d).$$



- Netz der konvexen Hülle besteht aus Dreiecken auf dem Paraboloid.
- Dreiecke induzieren Dreiecke in der Ebene.
- Ebene durch ein Dreieck auf dem Paraboloid schneidet eine „Kuppe“ ab, die keinen der projizierten gegebenen Punkte enthält.
- Projektion des Randes der Kuppe in die Ebene ist ein Kreis.
- Kreis ist Umkreis des Dreiecks in der Ebene und enthält keinen der gegebenen Punkte.

Algorithmus (d -dimensionale Delaunay-Triangulierung)**Eingabe:**

eine Punktmenge $\{\mathbf{p}_1, \dots, \mathbf{p}_n \in \mathbb{R}^d\}$.

Ausgabe:

eine d -dimensionale Delaunay-Triangulierung, dargestellt durch einen Zellinzidenzgraphen.

Ablauf:

BEGIN

berechne die konvexe Hülle von $T(\mathbf{p}_1), \dots, T(\mathbf{p}_n)$;

bestimme den Teilgraphen des Zellinzidenzgraphen, der die von der zur $(d+1)$ -Achse senkrechten Koordinatenebene aus sichtbaren Zellen beschreibt

END.

Bemerkungen:

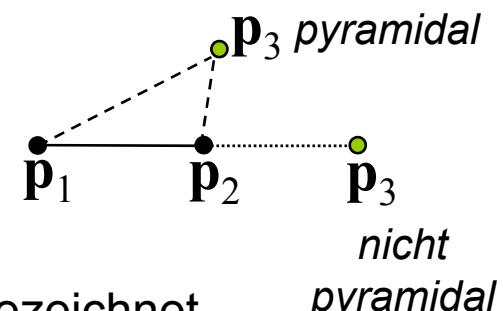
- *Lexikographisches Sortieren* bedeutet, dass zunächst nach der ersten Koordinate, bei gleicher erster Koordinate nach der zweiten Koordinate und so fort sortiert wird.
- Bei der Hinzunahme des nächsten Punktes \mathbf{p}_i sind zwei Fälle zu unterscheiden: \mathbf{p}_i kann in dem von den Punkten $\mathbf{p}_1, \dots, \mathbf{p}_{i-1}$ aufgespannten linearen Unterraum, deren *affiner Hülle* $\text{aff}\{\mathbf{p}_1, \dots, \mathbf{p}_{i-1}\}$, liegen oder nicht.

Affine Hülle einer Menge: der kleinste affine Unterraum (Punkt, Gerade, Ebene, ...), der die gegebene Menge enthält.

Beispiel:

Der dritte hinzugenommene Punkt \mathbf{p}_3 kann auf der Geraden durch \mathbf{p}_1 und \mathbf{p}_2 liegen oder nicht.

Die zweite Situation wird als *pyramidale Hinzunahme* bezeichnet.



Algorithmus (d -dimensionale konvexe Hülle)

Eingabe: n Punkte

Ausgabe: die konvexe Hülle von $\mathbf{q}_1, \dots, \mathbf{q}_n$

Ablauf:

BEGIN

bestimme die lexikographisch sortierte Anordnung $\mathbf{p}_1, \dots, \mathbf{p}_n$ der gegebenen Punkte \mathbf{q}_i ;

initialisiere die konvexe Hülle als leere Menge;

FOR $i := 1$ **TO** n **DO**

{ füge \mathbf{p}_i zur konvexen Hülle von $\mathbf{p}_1, \dots, \mathbf{p}_{i-1}$ hinzu }

END.

Algorithmus (Beneath-Beyond-Algorithmus, pyramidale Hinzunahme)

Eingabe: die konvexe Hülle zu $\mathbf{p}_1, \dots, \mathbf{p}_{i-1}, \mathbf{p}_i \notin \text{aff}\{\mathbf{p}_1, \dots, \mathbf{p}_{i-1}\}$.

Ausgabe: die konvexe Hülle von $\mathbf{p}_1, \dots, \mathbf{p}_i$.

Ablauf:

BEGIN

konstruiere die Punktmenge, deren Rand eine Pyramide ist, deren Grundfläche die bisher konstruierte konvexe Hülle ist und deren Spitze der neu hinzukommende Punkt ist;

gib die Punktmenge als Ergebnis aus;

END.

Algorithmus (Beneath-Beyond-Algorithmus, nichtpyramide Hinzunahme)

Eingabe: die konvexe Hülle zu $\mathbf{p}_1, \dots, \mathbf{p}_{i-1}$,
ein weiterer Punkt $\mathbf{p}_i \notin \text{aff}\{\mathbf{p}_1, \dots, \mathbf{p}_{i-1}\}$.

Ausgabe: die konvexe Hülle von $\mathbf{p}_1, \dots, \mathbf{p}_i$.

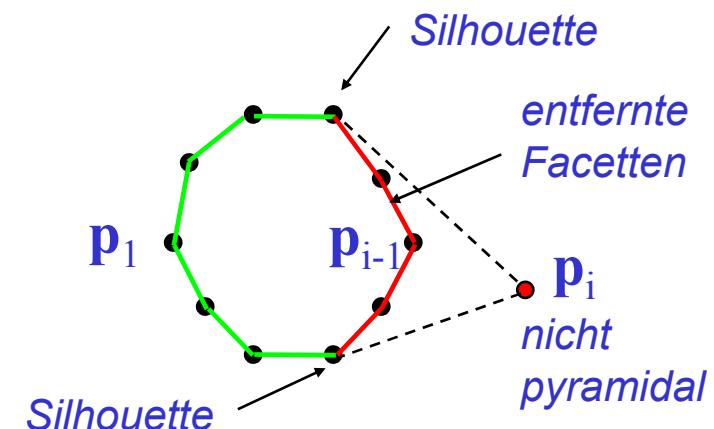
Ablauf:

BEGIN

entferne alle Facetten der bisher
konstruierten konvexen Hülle,
die von \mathbf{p}_i aus sichtbar sind;

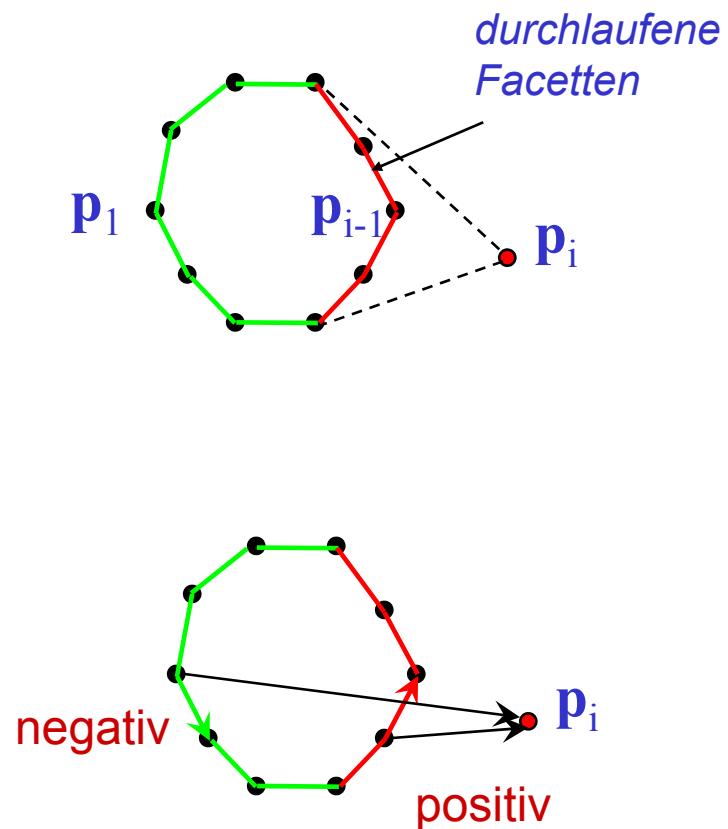
verbinde \mathbf{p}_i durch Facetten mit dem Rand der resultierenden Fläche, d.h.
den Silhouettenfacetten der bisher konstruierten konvexen Hülle

END.



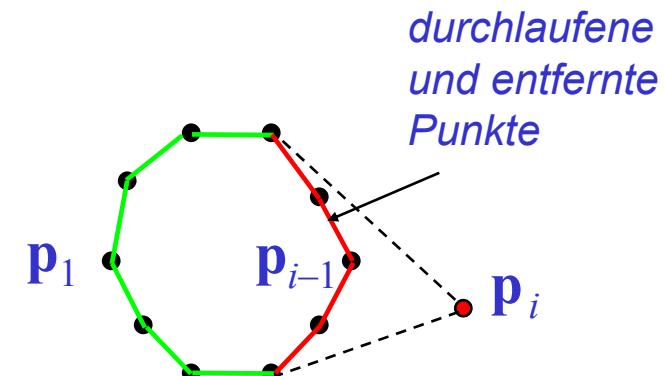
Bemerkungen:

- Die von p_i aus sichtbaren Flächen und die entsprechenden Silhouettenfacetten lassen sich durch Ablaufen der Oberfläche des Randes der konvexen Hülle ausgehend von p_{i-1} finden, da p_{i-1} von p_i aus sichtbar ist.
- Wenn die Normalenvektoren der volldimensionalen Facetten alle nach außen zeigen, lässt sich die Entscheidung durch das Vorzeichen eines Skalarprodukts treffen.



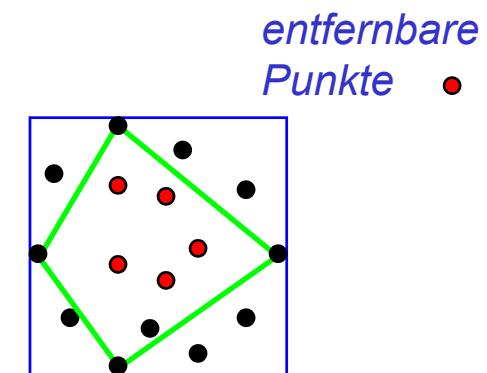
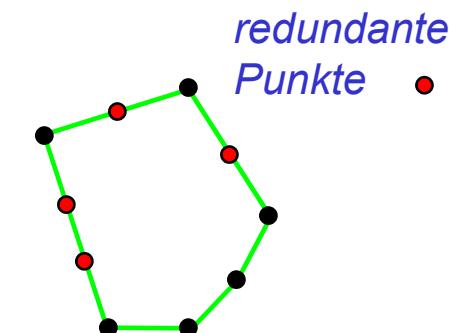
Bemerkungen:

- Für eine zweidimensionale Punktmenge gibt es eine Realisierung des Algorithmus mit $O(n \log n)$ Rechenzeit.
- Die Sortierphase hat einen Zeitaufwand von $O(n \log n)$.
- Die gesamte Hinzunahmephase hat einen Zeitaufwand von $O(n)$. Der Grund ist, dass die Rechenzeit proportional zur Anzahl der Punkte ist, die insgesamt beim Anfügen des nächsten Punktes inspiziert werden. Wenn die Inspektion ausgehend von p_{i-1} erfolgt, wird im Wesentlichen jeder inspizierte Punkt auch entfernt. Da nicht mehr als n Punkte existieren, ergibt sich der Zeitaufwand $O(n)$.



Bemerkungen:

- Der Algorithmus kann so realisiert werden, dass redundante Facetten, d.h. Facetten, die zu größeren Facetten zusammengefasst werden können, vermieden werden.
- Manchmal ist es lohnend, Vortests anzuwenden, mit denen unter Umständen Punkte im Inneren der konvexen Hülle entfernt werden können,
Bsp.: Punkte im Inneren der konvexen Hülle der Punkte der gegebenen Punktmenge, die auf dem achsenparallelen Hüllquader liegen.



Literatur:

- H. Edelsbrunner, Algorithms in Combinatorial Geometry, Springer-Verlag, 1987
- S. Abramowski, H. Müller, Geometrisches Modellieren, B.I.Wissenschaftsverlag, 1991, Kap. 8.3

E.1. Räumliche Daten

- E.1.1. Definition
- E.1.2. Repräsentationsformen
 - E.1.2.1. Punktbasierte Repräsentation
 - E.1.2.2. Voxelbasierte Repräsentation
 - E.1.2.3. Rechtecksgitter
 - E.1.2.4. Gekrümmte Gitter
 - E.1.2.5. Unstrukturierte (irreguläre) Gitter

E.2. Alpha-Shapes

- E.2.1. Einleitung
- E.2.2. Definitionen
- E.2.3. Alpha-Hülle
- E.2.4. Alpha-Form
- E.2.5. Randstreckung
- E.2.6. Bemerkungen
- E.2.7. Beispiel

E.3. Delaunay Triangulierung

- E.3.1. Zellkomplexe
 - E.3.1.1. Simplizes
 - E.3.1.2. Triangulierung
- E.3.2. Delaunay Triangulierung
 - E.3.2.1. Definitionen
 - E.3.2.2. Bemerkungen
 - E.3.2.3. Eigenschaften
 - E.3.2.4. Delaunay-Triangulierung/konvexe Hülle
 - E.3.2.5. d -dimensionale Delaunay-Triangulierung
 - E.3.2.6. Algorithmus „d-dimensionale konvexe Hülle“
 - E.3.2.7. Beneath-Beyond-Algorithmus
 - E.3.2.8. Bemerkungen

E.4. Netzreduktionsverfahren

- E.4.1. Elementanzahlreduktion
- E.4.2. Fehlermetriken

E.5. Volumenvisualisierung

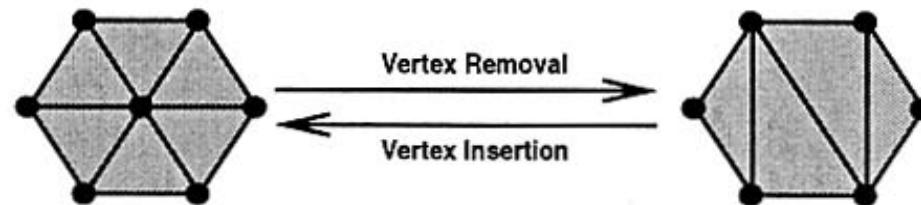
- E.5.1. Einführung
- E.5.2. Segmentierung
- E.5.3. Marching-Cubes-Algorithmus
 - E.5.3.1. Grundalgorithmus
 - E.5.3.2. Elementare Konfigurationen
 - E.5.3.3. Berechnung des Unterteilungspunkts
 - E.5.3.4. Bemerkungen
 - E.5.3.5. Adaptives Marching-Tetrahedron-Verfahren
 - E.5.3.6. Auflösung von Mehrdeutigkeiten

Inkrementelle Netzdezimierung:

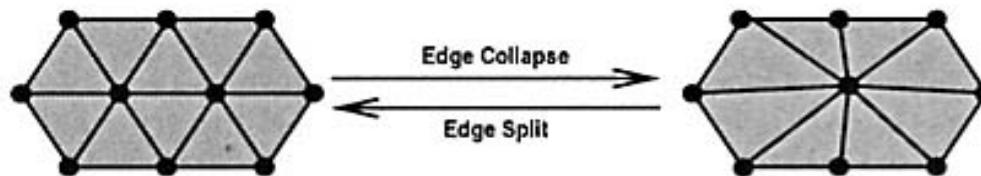
Anwendung einer Folge von kombinatorischen Operationen

Kombinatorische Operatoren:

- **Knotenentfernung** (vertex removal):

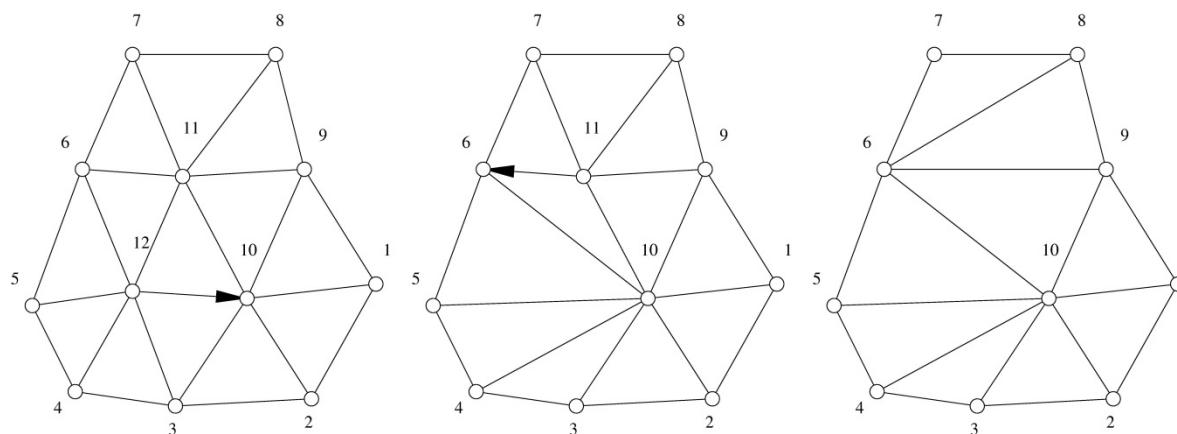


- **Kantenkontraktion** (edge contraction/collapse):

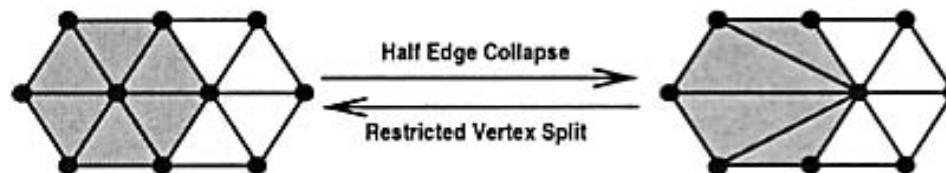


Bemerkung:

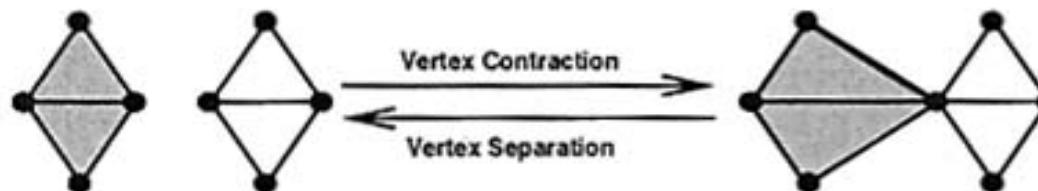
Der neue Punkt kann etwa als Mittel der zwei Endpunkte der zusammengezogenen Kante bestimmt werden.



- **Halbkantenkontraktion** (edge contraction/collapse):

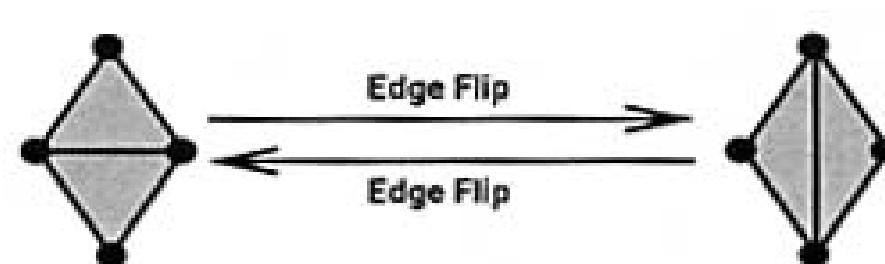


- **Knotenkontraktion** (vertex contraction):



Bemerkung: Die Knotenkontraktion kann die Netztopologie verändern.

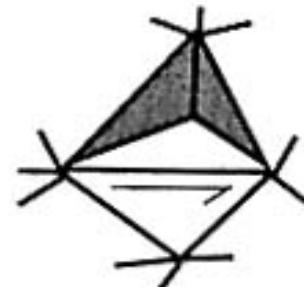
- **Kantenvertauschung**
(edge flipping):



Bemerkungen:

1. Zu den Operationen gibt es die jeweils inversen Operationen. Sie sind in den Abbildungen aufgeführt. Diese werden im Zusammenhang mit vielfach aufgelösten Netzen benötigt.
2. Bei der Anwendung der Operationen muss darauf geachtet werden, dass zu garantierende Eigenschaften des Netzes nicht verloren gehen. So verändert beispielsweise die Knotenkontraktion die Netztopologie.

Beispiel:



Durch Anwendung einer Halbkantenkontraktion in der gezeigten Weise ergeben sich Dreiecke, die in einer Ebene aufeinander liegen.

Algorithmus (Inkrementelle Netzdezimierung)

Eingabe:

ein Netz M , eine Fehlerschranke $\varepsilon > 0$ bezüglich einer Fehlermetrik.

Ausgabe:

ein Netz M' , das sich durch Dezimierung aus M ergibt und dessen Abweichung von M kleiner als ε ist.

Datenstrukturen:

eine Priority Queue Q für die anzuwendenden Operationen, sortiert nach aufsteigenden Kosten für die Operationen

Ablauf:

Für alle möglichen Gebiete G :

$op :=$ Operation auf dem Gebiet G ;

$c :=$ Kosten der Anwendung von op ;

Wenn $c < \varepsilon$ dann

füge (op, c) in Q ein;

Wiederhole:

Wende die Operation op_{opt} in Q mit den geringsten Kosten an;

Aktualisiere Q ;

bis Q leer ist;

gib das Ergebnisnetz als M' aus.

Bemerkungen:

1. Es wird eine Folge von Operationen angewandt, die das Netz lokal verändern. Ein Teil des Netzes, der durch eine Operation veränderbar ist, wird als mögliches Gebiet bezeichnet. Typische Operationen werden durch die unten aufgeführten kombinatorischen Operatoren definiert.
2. Ein Gebiet ist „möglich“, wenn sich durch Anwendung der entsprechenden Operation keine Verletzung von Netzeigenschaften ergibt, die erhalten werden müssen.
3. Von der Aktualisierung von Q sind alle Operationen betroffen, deren Gebiet sich durch die Anwendung von op_{opt} verändern. Ferner wird op_{opt} aus Q entfernt.

Fehlermetriken

Aspekte:

- gemessene Eigenschaften: Geometrie (Distanz, Ausgeglichenheit), Topologie, Attribute
- lokales bzw. globales Maß: Fehlerauswertung von Schritt zu Schritt oder Vergleich mit dem Originalnetz

Distanzmaße:

A. Dreiecksdistanz

B. Quadratische Fehlermetrik:

A. Dreieksdistanz

Bestimmung einer Distanz zwischen einem abgeleiteten Dreieck t und dem Teilnetz S des gegebenen Netzes, das in seine Berechnung eingegangen ist, für die Dreiecke t des reduzierten Netzes.

Approximative Lösungen:

- Fehlerakkumulation
- näherungsweise Hausdorff-Distanz

Fehlerakkumulation:

- **Voraussetzung:** Die Dreiecke des neuen Netzes ergeben sich durch Deformation von Dreiecken des alten Netzes.
- **Verfahren:** Für jedes Dreieck t des reduzierten Netzes Zuweisung einer Distanz von t zum Teilnetz S , aus dem t unmittelbar durch Anwendung einer Operation entstanden ist, und Addition dieser Distanz zum aktuellen Distanzwert von t vor der Reduktion.
- Beispiele für verwendbare Fehlermaße:
 - Maximale oder mittlere Distanz zwischen t und den Knoten von S
 - Maximale oder mittlere Distanz zwischen t und den Facetten von S
 - Anstelle einer skalaren Distanz können auch Distanzvektoren verwendet werden.

näherungsweise Hausdorff-Distanz:

- Distanz eines Punktes \mathbf{p} von einer Menge A :
$$d(\mathbf{p}, A) := \min \{d(\mathbf{p}, \mathbf{q}) \mid \mathbf{q} \in A\}$$
- einseitige *Hausdorff-Distanz* zwischen zwei Mengen A und B :
$$h(A, B) := \max \{d(\mathbf{p}, B) \mid \mathbf{p} \in A\}$$

Bemerkung: Im Allgemeinen ist $h(A, B) \neq h(B, A)$.
- *Hausdorff-Distanz* zwischen zwei Mengen A und B :
$$H(A, B) := \max \{h(A, B), h(B, A)\}$$

Verfahren:

Näherungsweise Berechnung der einseitigen Hausdorff-Distanz $h(M, M')$, wobei M das Originalnetz und M' das aktuelle reduzierte Netz ist, nach folgendem Verfahren:

- Bei jeder Anwendung einer Reduktionsoperation Zuweisung der entfernten Knoten an das nächstliegende neue Dreieck.
- Falls sich Dreiecke verändern, neue Zuweisung der ihnen zugeordneten Knoten zu den dann nächsten Dreiecken.
- Berechnung der einseitigen Hausdorff-Distanz durch Bestimmung des am weitesten von einem Dreieck entfernten zugewiesenen Punktes.

B. Quadratische Fehlermetrik

- **Voraussetzung:** Das Netzreduktionsverfahren generiert neue Knoten. Deren Abstand zum alten Netz dient als Fehlermaß.
- **Definition:**

Zuweisung einer Fehlermetrik $E_p(\mathbf{x})$ an jeden Knoten p :

- falls p ein Knoten des Originalnetzes:

$$E_p(\mathbf{x}) := \sum_{s \in S} (\mathbf{n}_s^* \mathbf{x} - d_s)^2 / |S|$$

wobei S die Menge der Dreiecke, die zu p inzident sind,

\mathbf{n}_s der normierte Normalenvektor des Dreiecks s in S ,

$d_s := \mathbf{n}_s^* \mathbf{q}_s$, \mathbf{q}_s ein Eckpunkt von s und \mathbf{x} ein beliebiger Punkt.

- falls \mathbf{p} ein abgeleiteter Knoten: $E_{\mathbf{p}}(\mathbf{x}) := \sum_{\mathbf{q} \in Q} E_{\mathbf{q}}(\mathbf{x}) / |Q|$
wobei Q die Menge aller Knoten, die unmittelbar in die Berechnung von \mathbf{p} eingehen.
- **Verfahren:** Bestimmung der Kosten einer Operation, die einen Knoten \mathbf{p} generiert, durch $E_{\mathbf{p}}(\mathbf{p})$.

Bemerkung:

1. Der Fehler eines Punktes ergibt sich als Summe der Distanzen des Punktes von den Ebenen durch die Menge von Dreiecken, bezüglich der der Fehler bestimmt wird.
 2. Aufgrund der Mittelwertbildung liefert die quadratische Fehlermetrik weder eine obere noch eine untere Schranke des geometrischen Fehlers.
-

E.1. Räumliche Daten

- E.1.1. Definition
- E.1.2. Repräsentationsformen
 - E.1.2.1. Punktbasierte Repräsentation
 - E.1.2.2. Voxelbasierte Repräsentation
 - E.1.2.3. Rechtecksgitter
 - E.1.2.4. Gekrümmte Gitter
 - E.1.2.5. Unstrukturierte (irreguläre) Gitter

E.2. Alpha-Shapes

- E.2.1. Einleitung
- E.2.2. Definitionen
- E.2.3. Alpha-Hülle
- E.2.4. Alpha-Form
- E.2.5. Randstreckung
- E.2.6. Bemerkungen
- E.2.7. Beispiel

E.3. Delaunay Triangulierung

- E.3.1. Zellkomplexe
 - E.3.1.1. Simplizes
 - E.3.1.2. Triangulierung
- E.3.2. Delaunay Triangulierung
 - E.3.2.1. Definitionen
 - E.3.2.2. Bemerkungen
 - E.3.2.3. Eigenschaften
 - E.3.2.4. Delaunay-Triangulierung/konvexe Hülle
 - E.3.2.5. d -dimensionale Delaunay-Triangulierung
 - E.3.2.6. Algorithmus „d-dimensionale konvexe Hülle“
 - E.3.2.7. Beneath-Beyond-Algorithmus
 - E.3.2.8. Bemerkungen

E.4. Netzreduktionsverfahren

- E.4.1. Elementanzahlreduktion
- E.4.2. Fehlermetriken

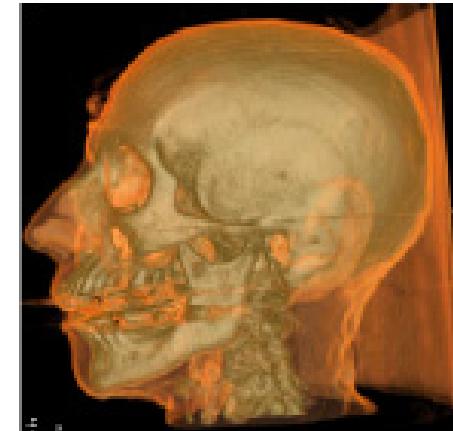
E.5. Volumenvisualisierung

- E.5.1. Einführung
- E.5.2. Segmentierung
- E.5.3. Marching-Cubes-Algorithmus
 - E.5.3.1. Grundalgorithmus
 - E.5.3.2. Elementare Konfigurationen
 - E.5.3.3. Berechnung des Unterteilungspunkts
 - E.5.3.4. Bemerkungen
 - E.5.3.5. Adaptives Marching-Tetrahedron-Verfahren
 - E.5.3.6. Auflösung von Mehrdeutigkeiten

Ansätze der Volumenvisualisierung:

- ***direkte Visualisierung:***

graphische Darstellung durch Verfahren,
die direkt die Repräsentation des
Volumens verwenden

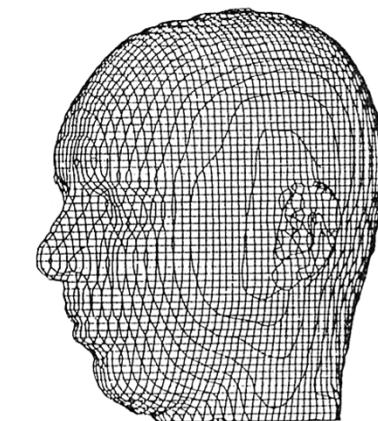


- ***Visualisierung durch geometrische Elemente:***

Berechnung von Trennflächen zwischen
Untervolumina mit vorgegebenen
Eigenschaften und graphische Darstellung
der Trennflächen

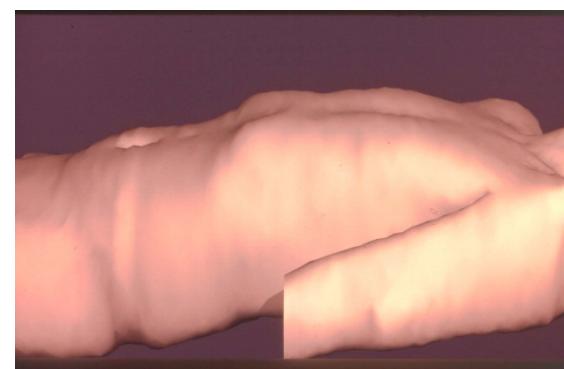
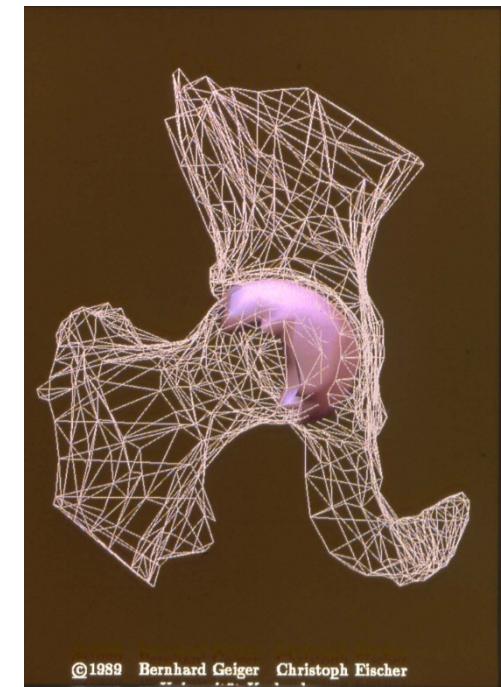
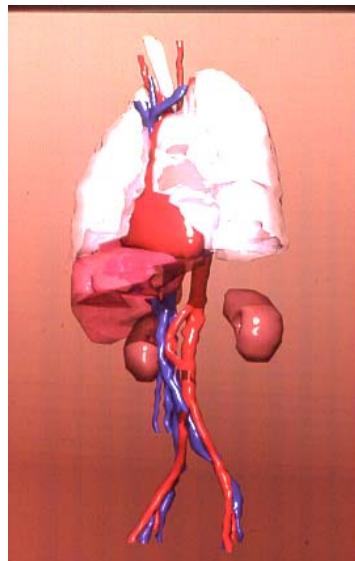
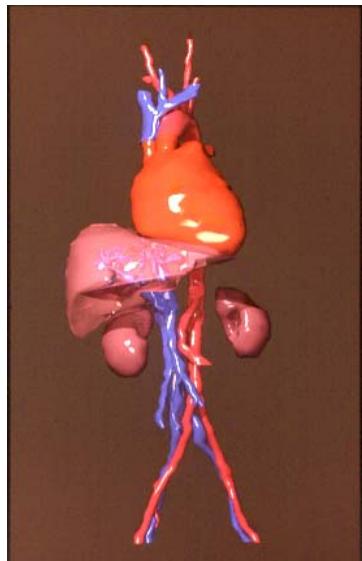
wesentliche Aufgaben:

- Segmentierung: Bestimmung der Untervolumina
- Berechnung der Trennflächen



Beispiel:

Visualisierung durch geometrische Elemente durch Oberflächendarstellung



Binäre Segmentierungsfunktion

$$b : \mathbb{R}^m \rightarrow \{0, 1\}$$

wobei ein Punkt $\mathbf{v} \in S$ als zu einem Objekt gehörend definiert wird, wenn $b(\mathbf{f}(\mathbf{v})) = 1$, und als zum Hintergrund zugehörig definiert wird, wenn $b(\mathbf{f}(\mathbf{v})) = 0$.

Trenn- oder Randfläche

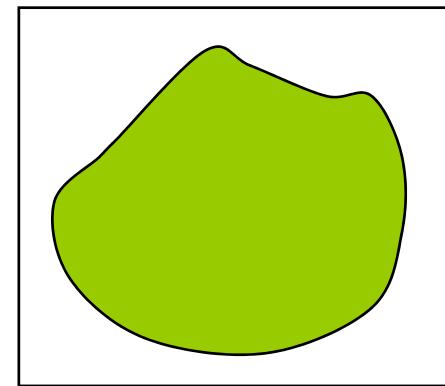
besteht aus allen Punkten, für die in jeder Umgebung Punkte \mathbf{o} mit $b(\mathbf{f}(\mathbf{o})) = 1$ und Punkte \mathbf{h} mit $b(\mathbf{f}(\mathbf{h})) = 0$, d.h. sowohl Objektpunkte als auch Hintergrundpunkte, liegen.

speziell für reellwertige Funktionen $f : \mathbb{R}^d \rightarrow \mathbb{R}$.

Schwellwertsegmentierungsfunktion

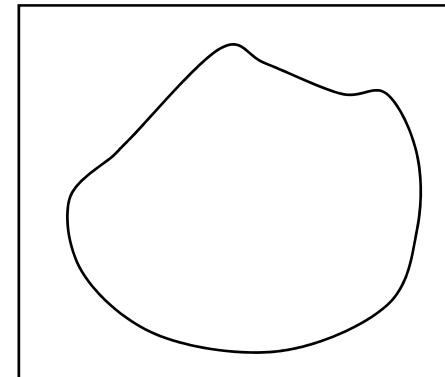
$$b : \mathbb{R} \rightarrow \{0, 1\}$$

mit $b(f) = 1$ genau dann, wenn $f \geq f_{\text{iso}}$,
 f_{iso} eine gegebene Konstante (Schwellwert),



Isoflächensegmentierung

$$S(f_{\text{iso}}) := \{\mathbf{v} \mid f(\mathbf{v}) = f_{\text{iso}}\},$$



Bemerkung

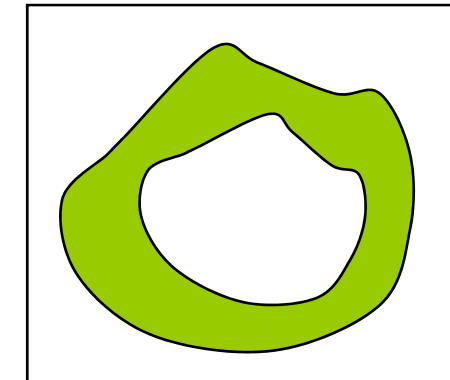
Iso-Flächen entsprechen Iso- oder Höhenlinien bei zweidimensionalen Funktionen.

Intervallsegmentierung

$$b : \mathbb{R} \rightarrow \{0, 1\}$$

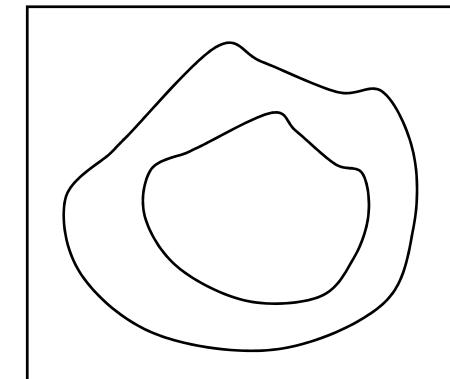
mit

$b(f) = 1$ genau dann, wenn $f_1 \leq f \leq f_2$,
 f_1, f_2 gegebene Konstanten



Iso-Kontur-Segmentierung

$$C(f_1, f_2) := \{\mathbf{v} \mid f(\mathbf{v}) \in \{f_1, f_2\}\}$$



Bemerkung

Die folgende Darstellung basiert im Wesentlichen auf der Repräsentation durch Voxelvolumen.

Berechnung einer Trennfläche für Voxelobjekte:

Marching-Cubes-Verfahren:

Eingabe:

Voxelvolumen und Objekt, das durch eine Segmentierungsfunktion definiert ist.

Ausgabe:

Ein Netz, das die Trennfläche des Objekts approximiert

BEGIN

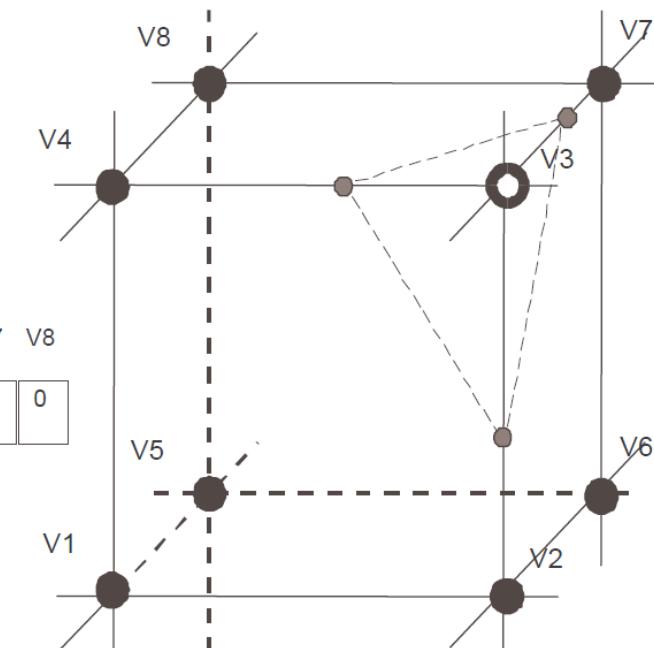
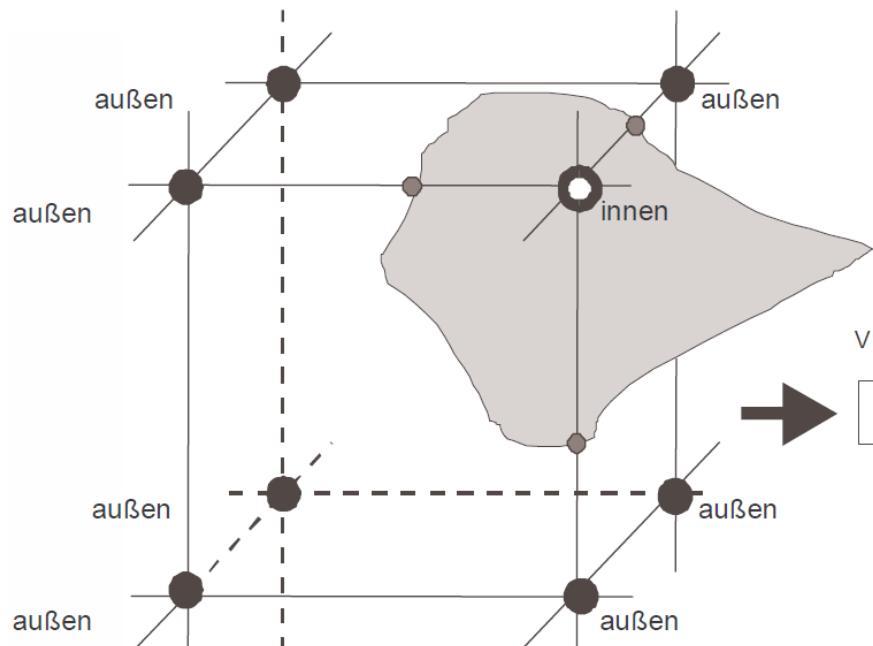
FOR alle Würfel des Modells **DO**

finde aufgrund der Zugehörigkeit der Eckpunkte zum Objekt die passende elementare Konfiguration und gib deren Facetten passend transformiert aus

END.

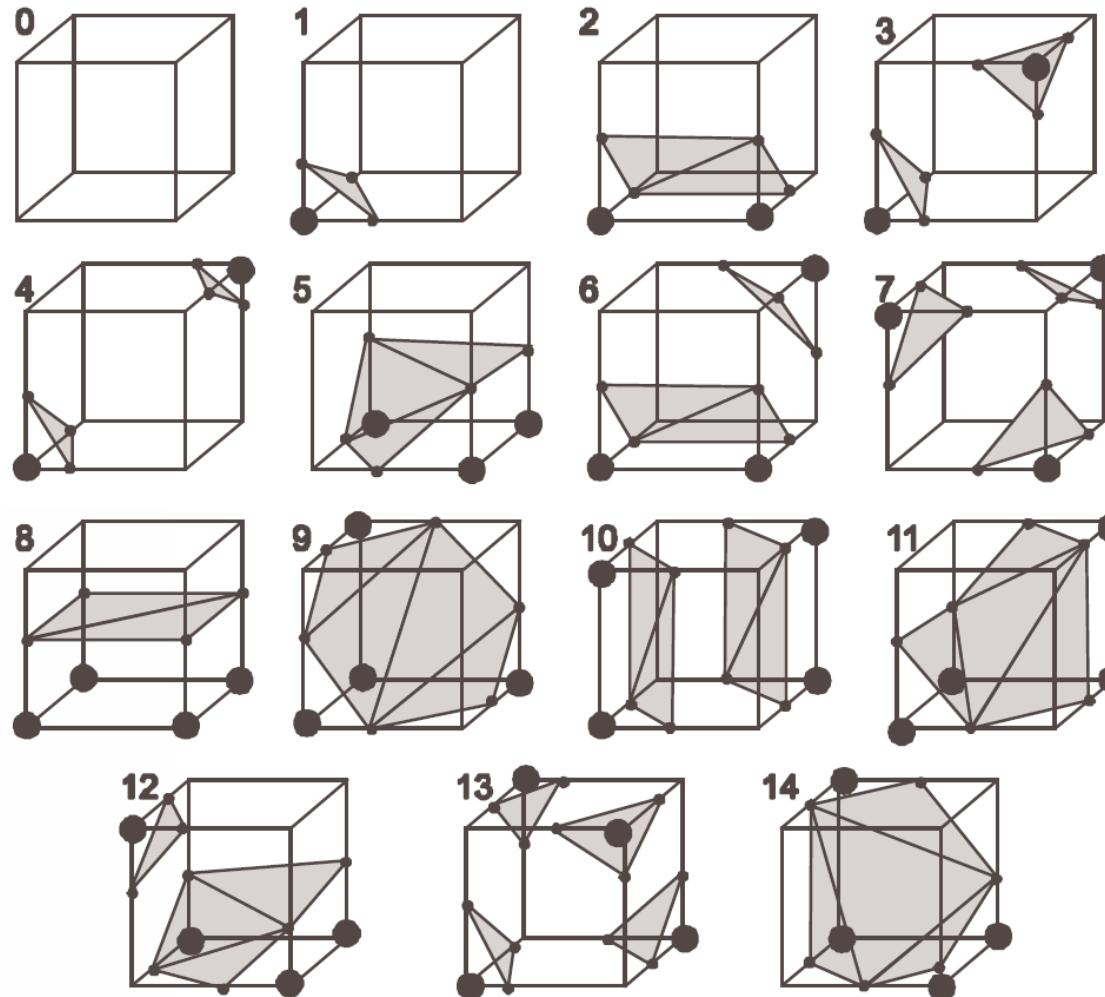
Idee des Marching-Cubes-Verfahren:

Ein Würfel (Quader) wandert im Datenwürfel von Zelle zu Zelle. Die Eckpunkte der Zellen werden gemäß der Schwelle klassifiziert (innen - außen). Die Schnittpunkte Isofläche/Würfelkante werden durch lineare Interpolation ermittelt und zu Flächen verbunden.



Elementare Konfigurationen

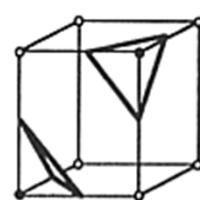
● Knoten im Voxelobjekt



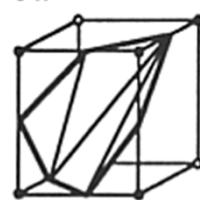
→ Die Bitzerlegung im Klassifizierungsvektor bestimmt das Flächenbild

Elementare Konfigurationen Fälle 3, 6, 7, 10, 12 und 13 nicht eindeutig

3



3a



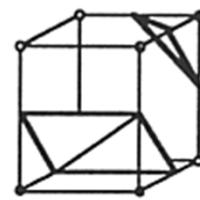
4



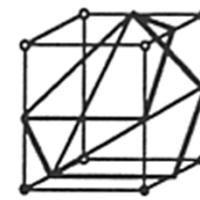
4a



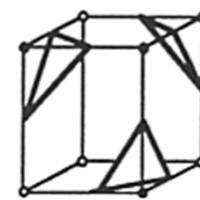
6



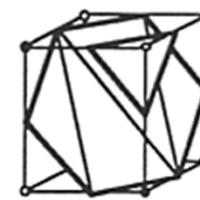
6a



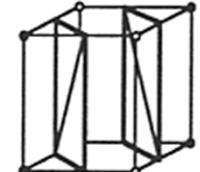
7



7a



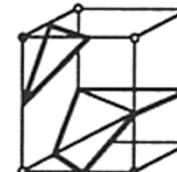
10



10a



12



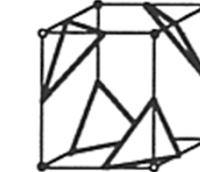
12a



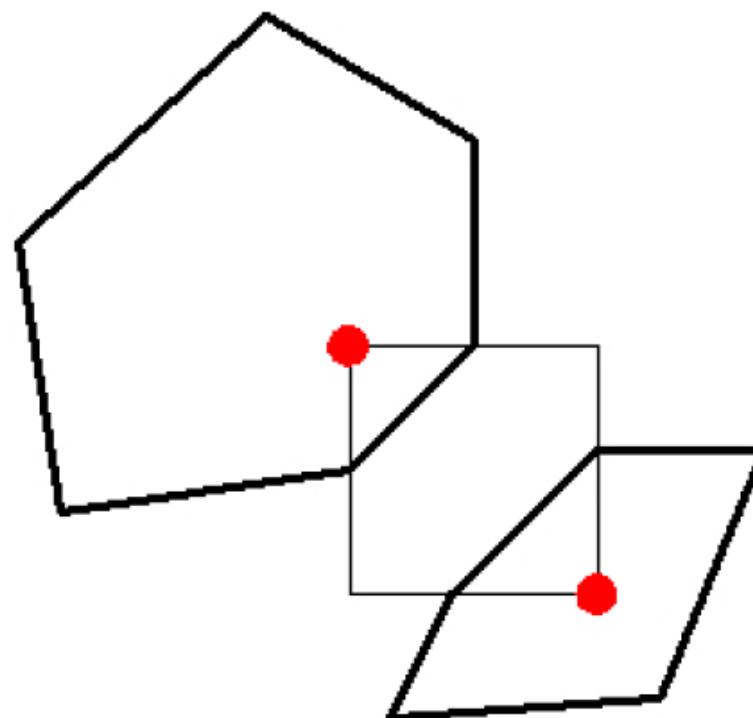
13



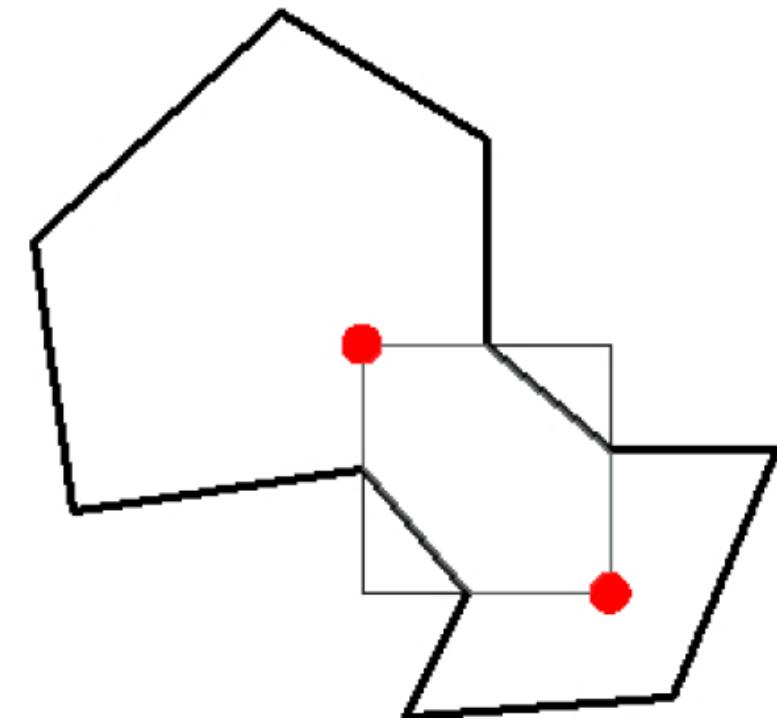
13a



Beispiel für einen nicht eindeutigen Fall:



Variante A

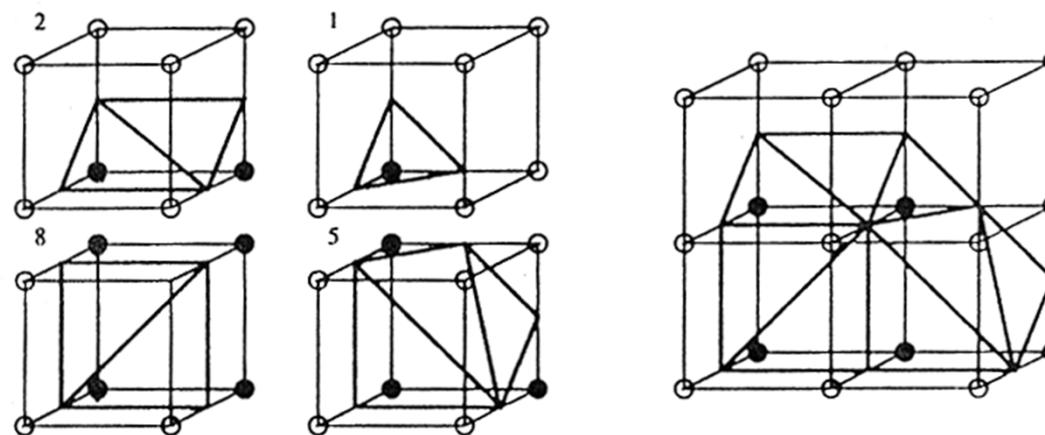


Variante B

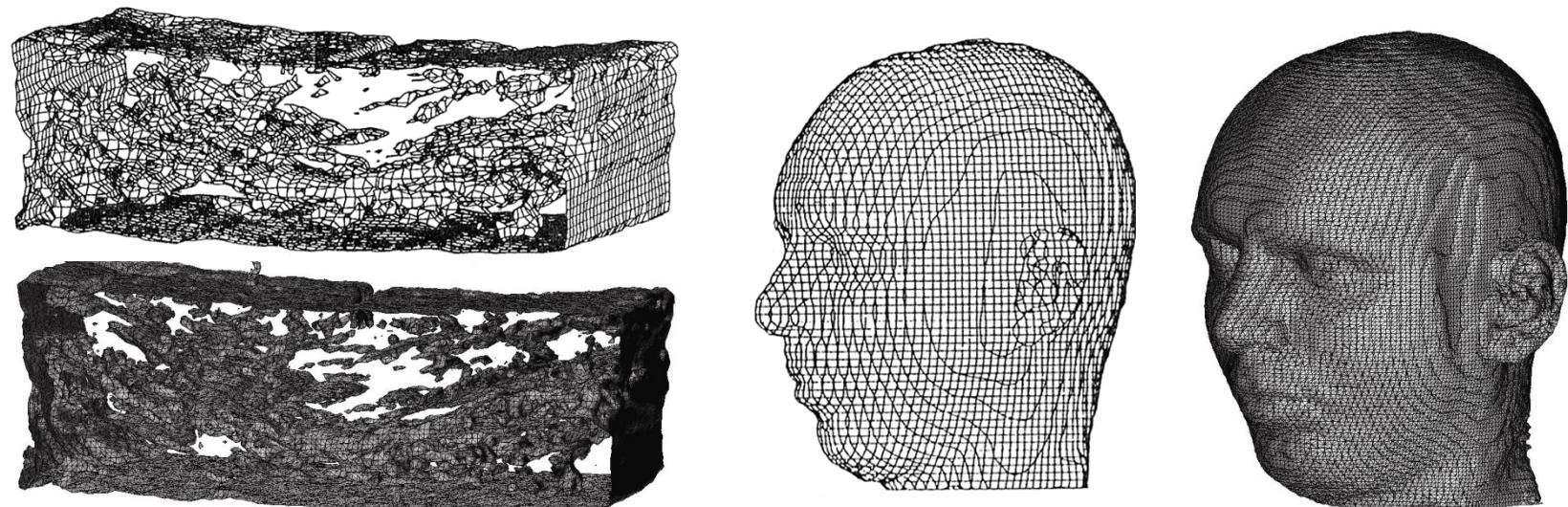
E.5.3. Marching-Cubes-Algorithmus

E.5.3.2. Elementare Konfigurationen

Beispiel 1:



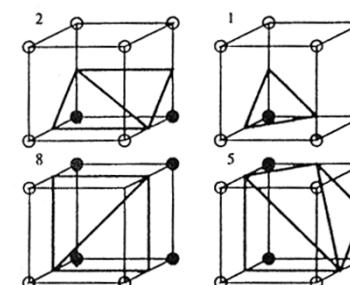
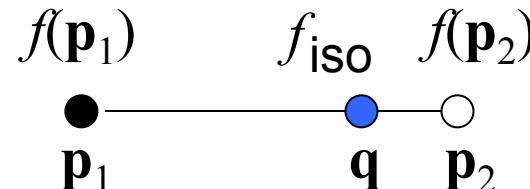
Beispiel 2:



Berechnung des Unterteilungspunkts q einer Kante p_1, p_2 im Fall der Iso-Flächen-Segmentierung $S(f_{\text{iso}}) := \{v \mid f(v) = f_{\text{iso}}\}$:

$$q = (1-\lambda) p_1 + \lambda p_2,$$

wobei λ die Lösung der Gleichung $(1-\lambda)f(p_1) + \lambda f(p_2) = f_{\text{iso}}$



Bemerkung:

Das Marching-Cubes-Verfahren lässt sich auf Gitter beliebiger Dimension übertragen.

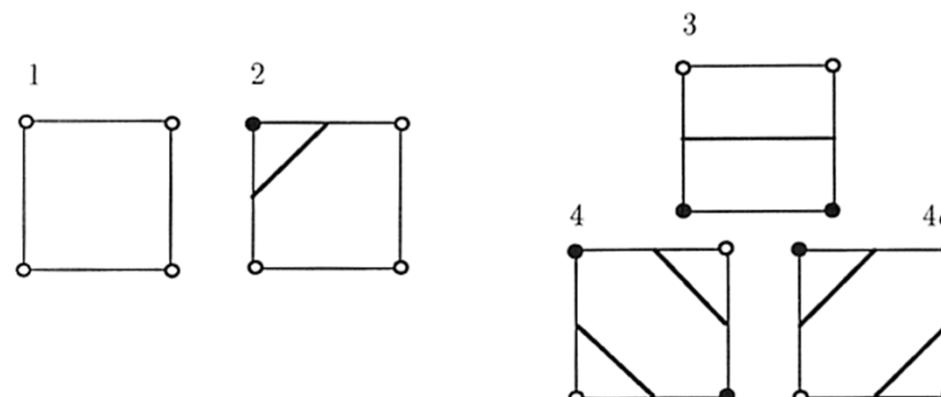
Literatur

M. Stark, Konturapproximation in mehrdimensionalen regulären Gitterdaten, Dissertation Universität Dortmund, FB Informatik, 1996

Beispiel:

2D: Marching-Squares-Verfahren

Konfigurationen:



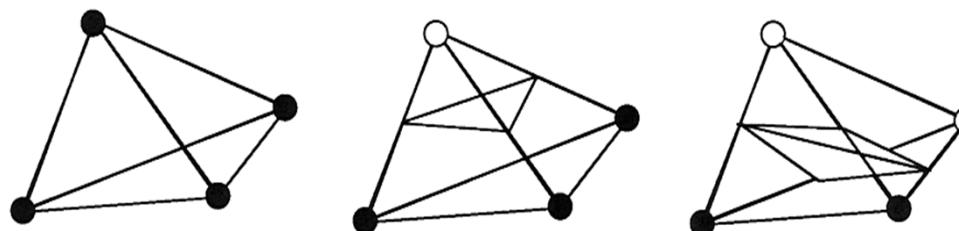
Bemerkung:

Der Ansatz der Marching-Cubes-Verfahrens lässt sich auf andere Gittertypen übertragen.

Beispiel:

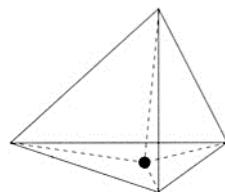
Dreiecks- und Tetraedernetze

Konfigurationen:

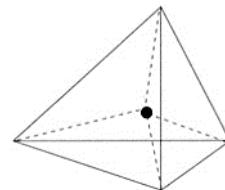


Vorteil: keine Mehrdeutigkeiten

- Approximationsbedingung auf einer Seite verletzt (iteriert anwenden, wenn auf mehreren Seiten verletzt):



- Approximationsbedingung im Inneren verletzt:



Literatur

D. Ruprecht, H. Müller, A scheme for edge-based adaptive tetrahedron subdivision, In: Hege, Polthier, Mathematical Visualization, Springer-Velrag, 1998, p. 61-70

Adaptives Marching-Tetrahedron-Verfahren

Eingabe:

Eine reellwertige Funktion f mit einem dreidimensionalen Definitionsbereich D , wobei D durch ein (grobes) Tetraedernetz gegeben ist, $\varepsilon > 0$.

Ausgabe:

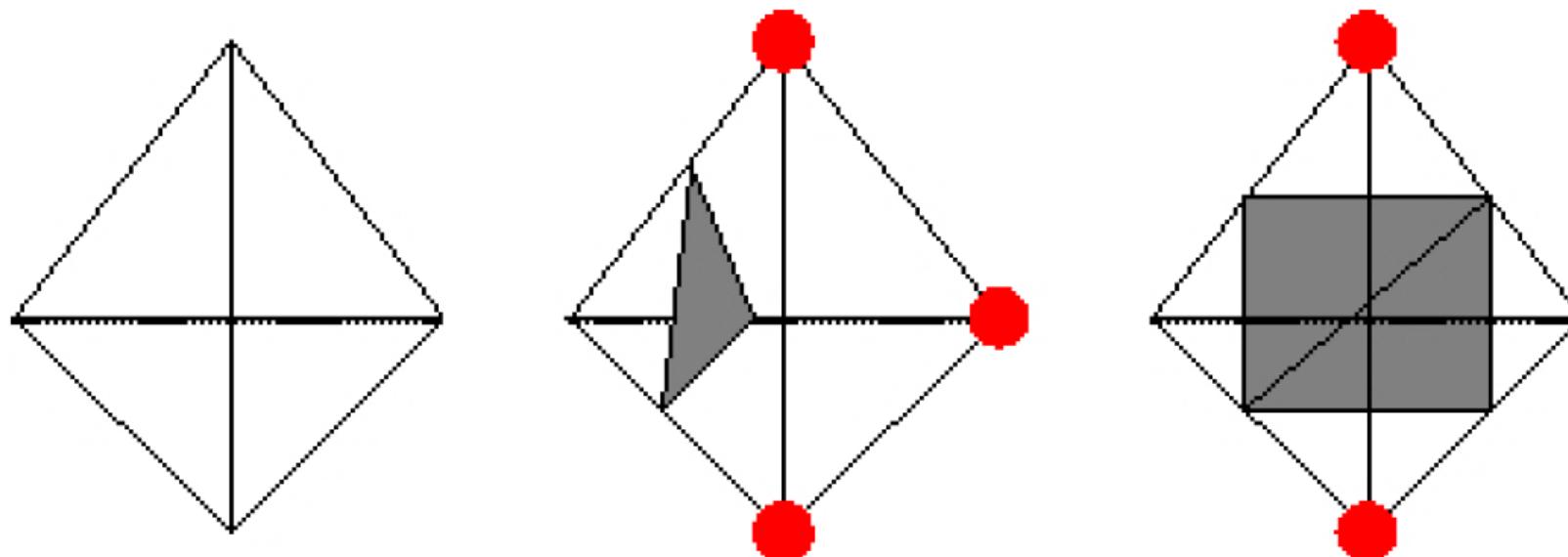
Eine adaptive Verfeinerung von D , sodass die betragsmäßige Differenz zwischen f und der Funktion, die sich durch trilineare Interpolation der Funktionswerte an den Eckpunkten eines Tetraeders in sein Inneres ergibt, kleiner als ε ist.

Verfahren:

Initialisiere eine Liste L der noch zu bearbeitenden Tetraeder mit den Tetraedern des gegebenen Netzes D , auf denen der Approximationsfehler größer als ε ist (Approximationsbedingung nicht erfüllt).

Adaptives Marching-Tetrahedron-Verfahren

Die 16 möglichen Fälle für den lokalen Schnitt der Fläche mit einem Tetraeder lassen sich auf durch Reduktion auf 3 Fälle begrenzen.



Bemerkung:

Obwohl resultierend mehr Dreiecke resultieren, ist die resultierende Triangulierung der Oberfläche in der Regel besser.

Für alle Tetraeder t in L :

{ Falls t Kanten hat, die die Approximationsbedingung nicht erfüllen, dann

{ wähle je einen Unterteilungspunkt auf den entsprechenden Kanten;
ersetze t durch Untertetraeder der entsprechenden Konfiguration;

}

Sonst:

Falls t Seitenflächen hat, die die Approximationsbedingung nicht
erfüllen, dann

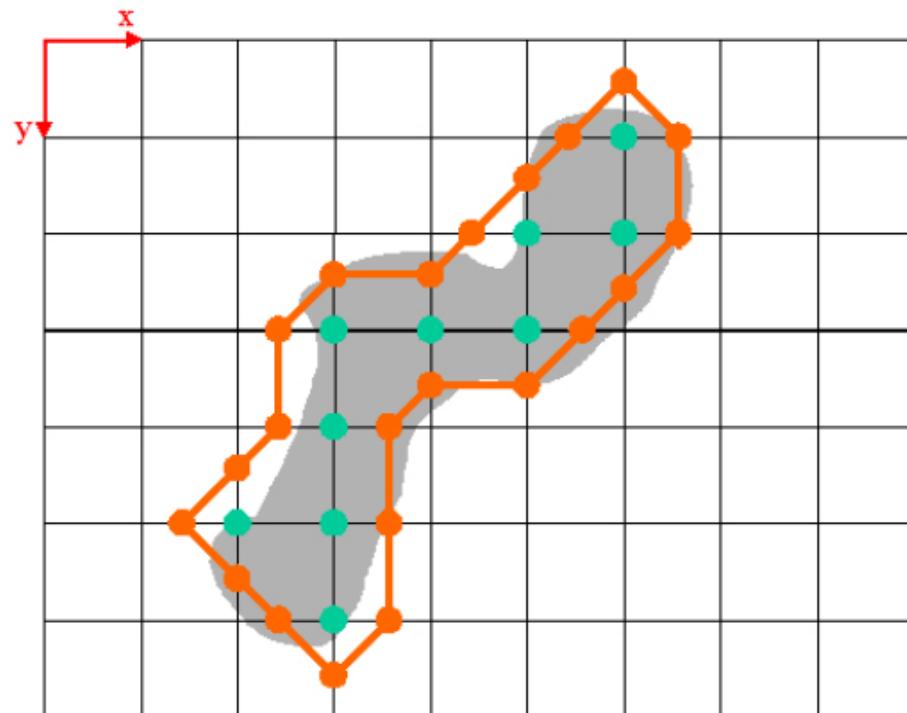
ersetze t durch Untertetraeder der entsprechenden Konfiguration;

Sonst: ersetze t durch die Untertetraeder der Konfiguration zum Fall,
dass das Innere von t die Approximationsbedingung nicht erfüllt;

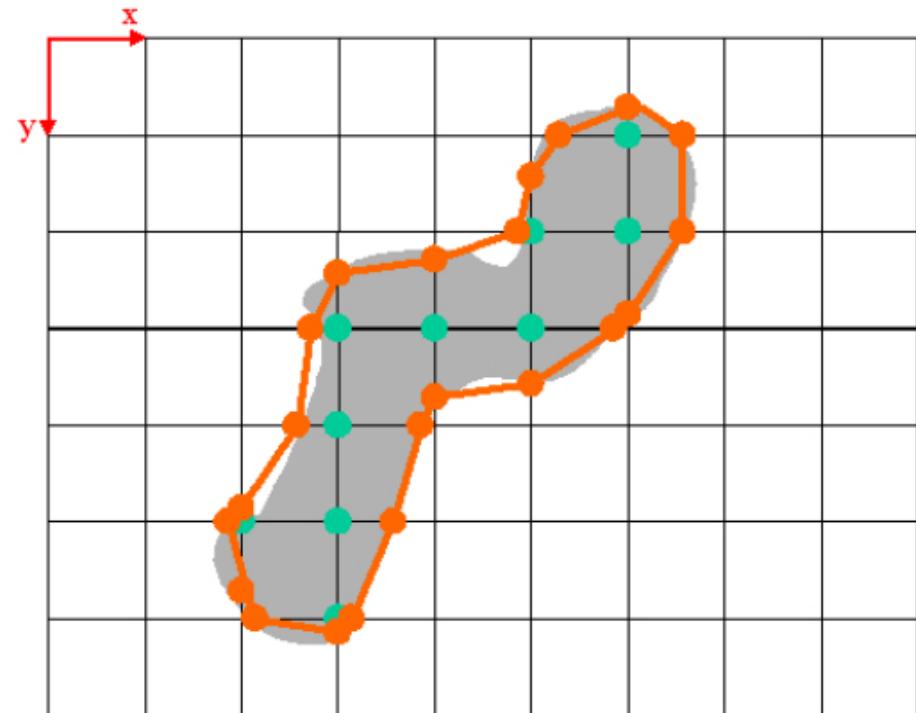
}

füge die Untertetraeder, die die Approximationsbedingung nicht
erfüllen, an L an und gib die anderen als Teil des Ergebnisses aus;

Beispiel (2D-Fall):



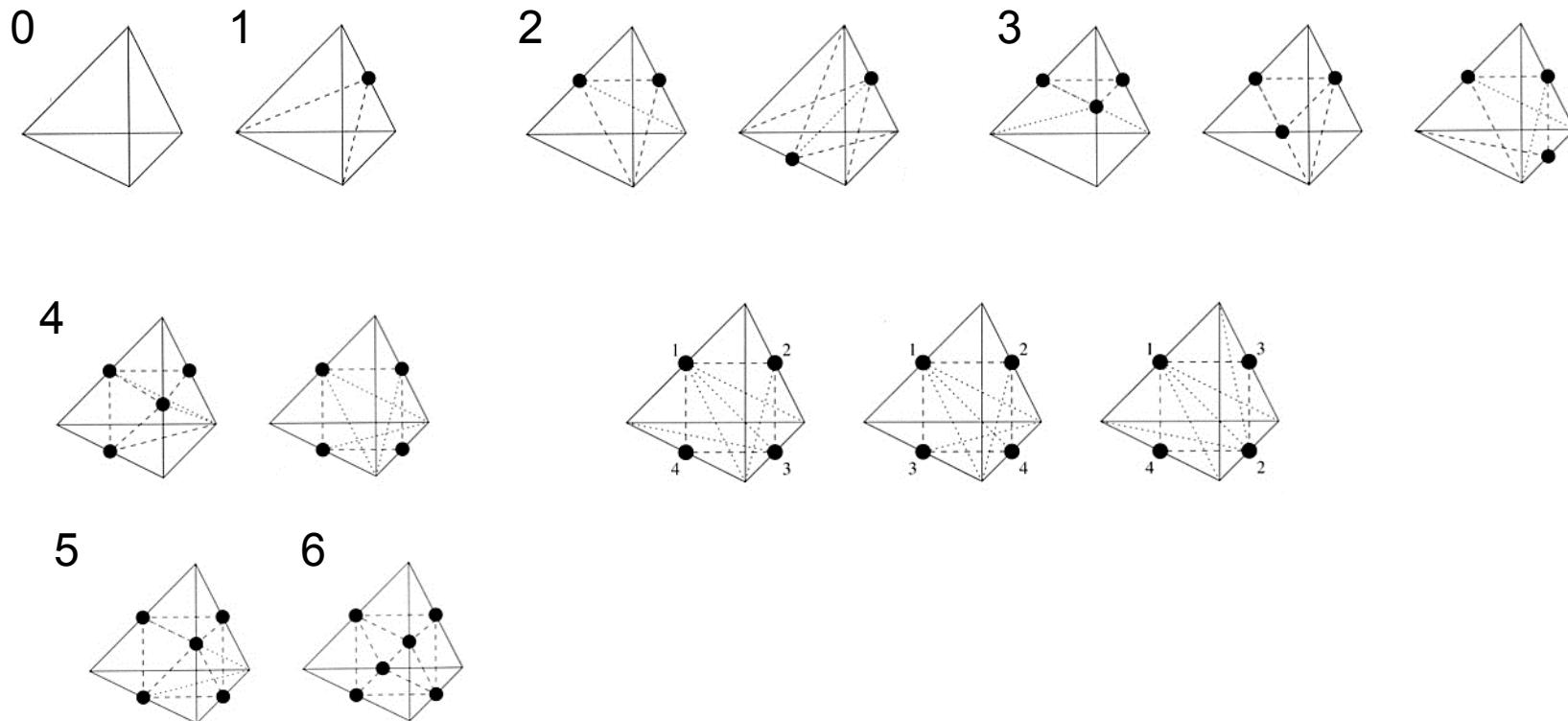
Standardversion



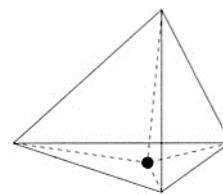
Adaptive Version

Tetraederkonfigurationen:

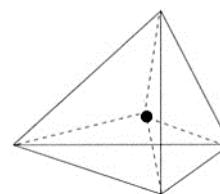
- Approximationsbedingung auf Kanten verletzt:



- Approximationsbedingung auf einer Seite verletzt (iteriert anwenden, wenn auf mehreren Seiten verletzt):



- Approximationsbedingung im Inneren verletzt:



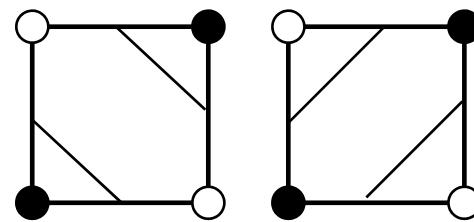
Literatur:

D. Ruprecht, H. Müller, A scheme for edge-based adaptive tetrahedron subdivision, In: Hege, Polthier, Mathematical Visualization, Springer-Verlag, 1998, p. 61-70

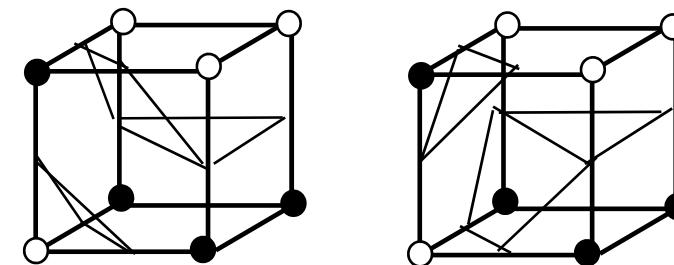
Problem der mehrdeutigen Konfigurationen:

Beispiel:

zweidimensional (Marching Squares)



dreidimensional



Lösungsansatz:

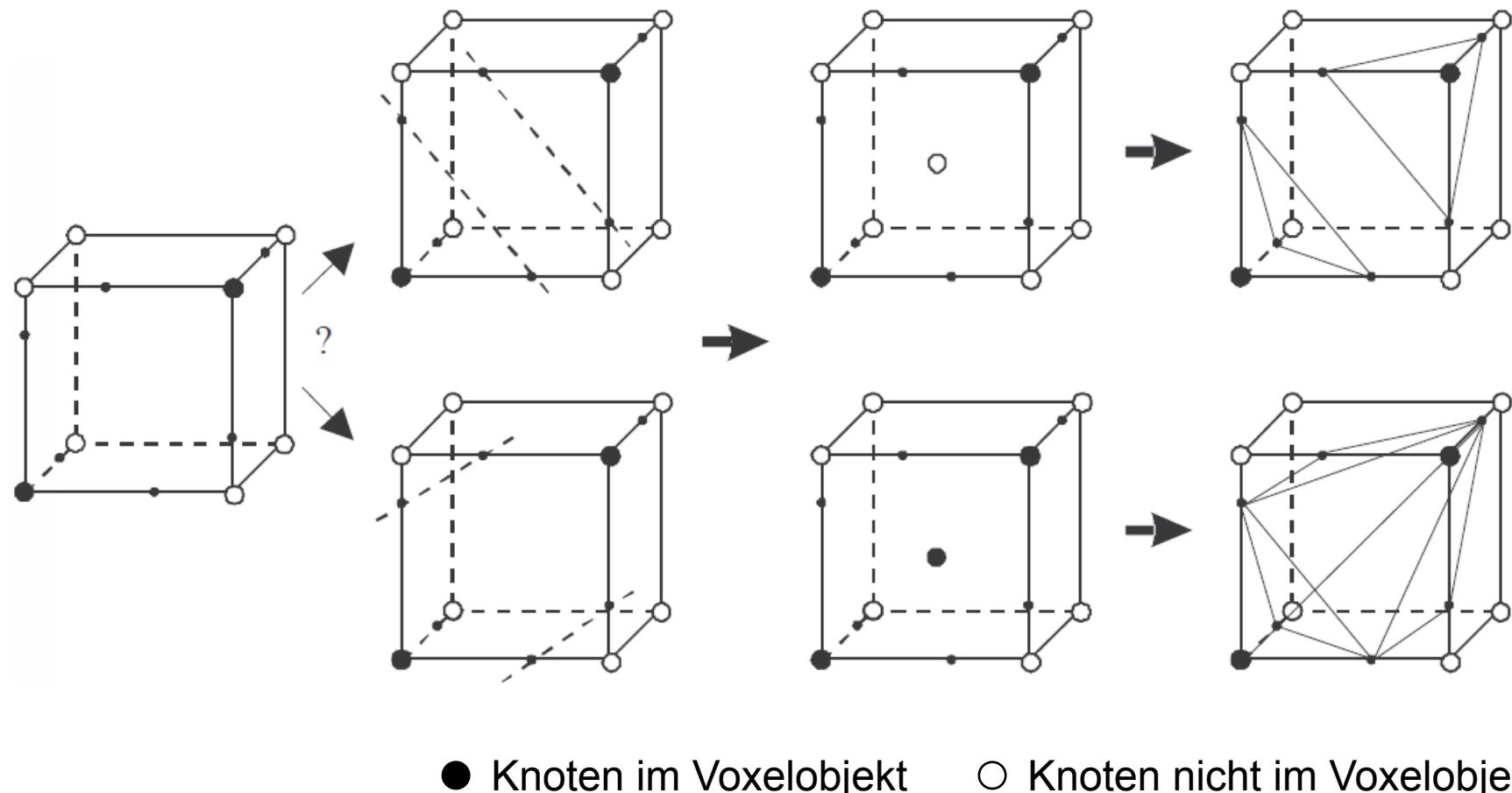
Interpoliere die Daten an den Eckpunkten und analysiere das Verhalten der Iso-Fläche der interpolierenden Funktion.

Bekanntes Verfahren:

Asymptotic Decider (Asymptotische Entscheidungsregel)

Auflösen der Mehrdeutigkeit

Ein weiterer Datenpunkt in der Mitte der Fläche wird durch Klassifikation ermittelt und auch klassifiziert



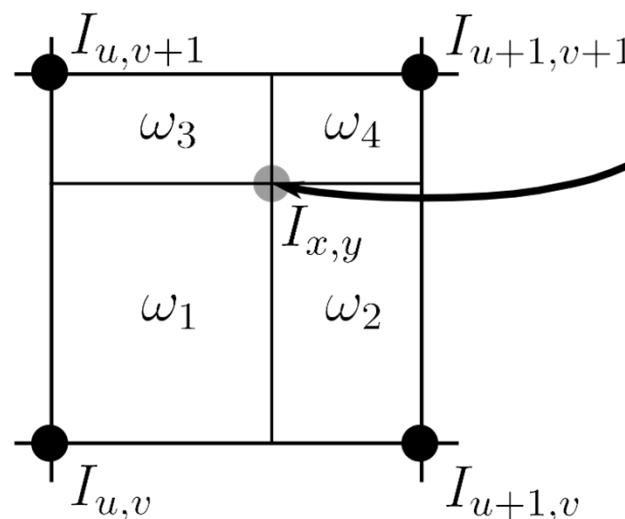
Asymptotische Entscheidungsregel (asymptotic decider):

Auflösung der Mehrdeutigkeiten im zweidimensionalen Fall durch bilineare Interpolation.

Bilineare Interpolation:

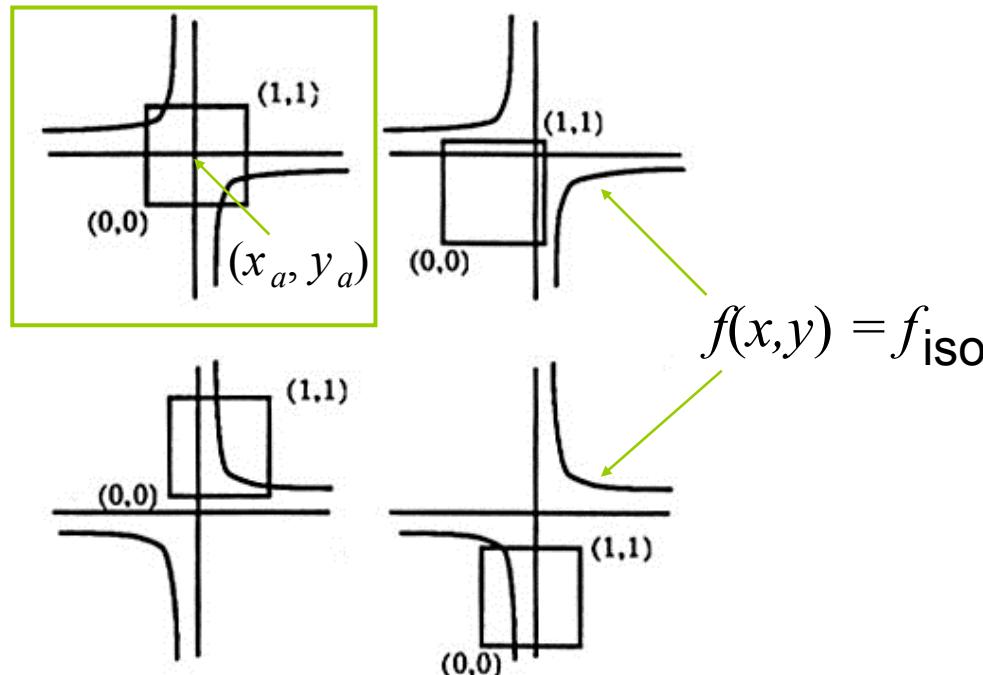
$$\begin{aligned} f(x, y) = & (1 - y) \cdot ((1 - x) \cdot f(0, 0) + x \cdot f(1, 0)) \\ & + y \cdot ((1 - x) \cdot f(0, 1) + x \cdot f(1, 1)) \end{aligned}$$

für Gitterschrittweite gleich 1 in alle Richtungen



Mögliche Fälle für $f(x,y) = f_{\text{ISO}}, f(x,y)$ bilineare Interpolation:

Fall mit
potentieller
Mehrdeutigkeit



Bemerkung:

$f(x,y) = f_{\text{ISO}}$ definiert eine Kegelschnittkurve, speziell eine Hyperbel.