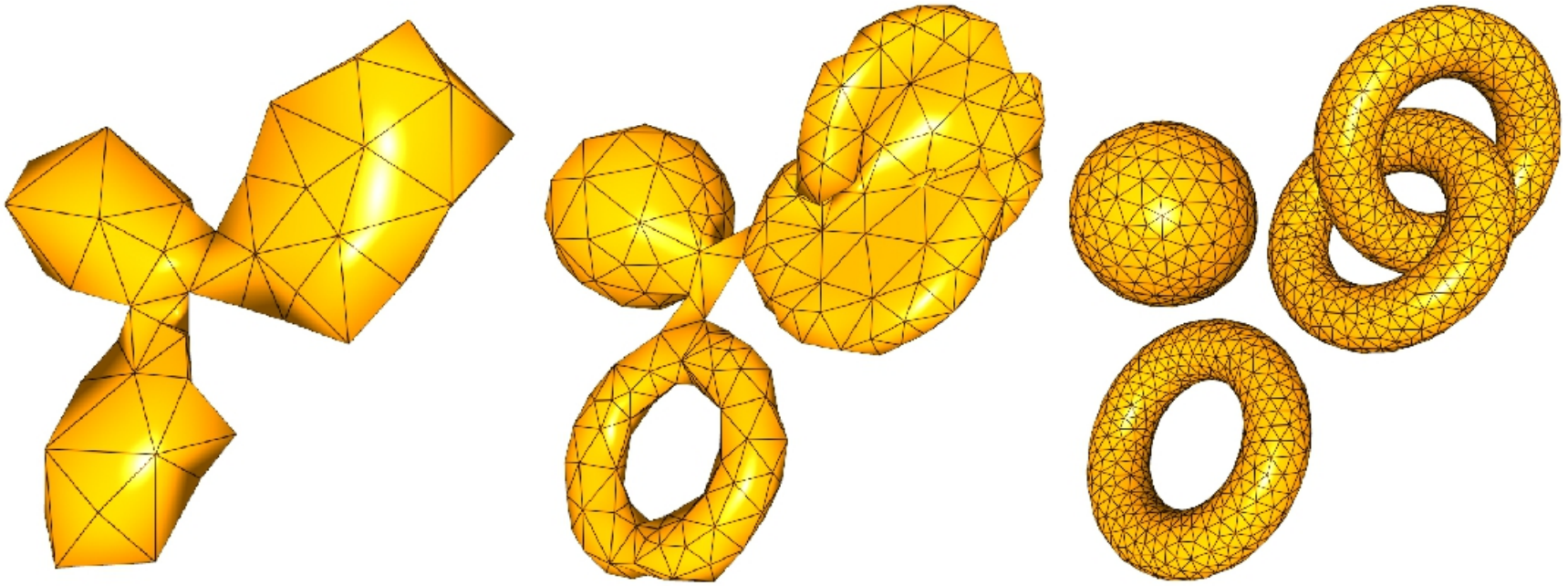
Objektrekonstruktion durch Smart Growing Cells

Henrik Annuth & Christian-A. Bohn



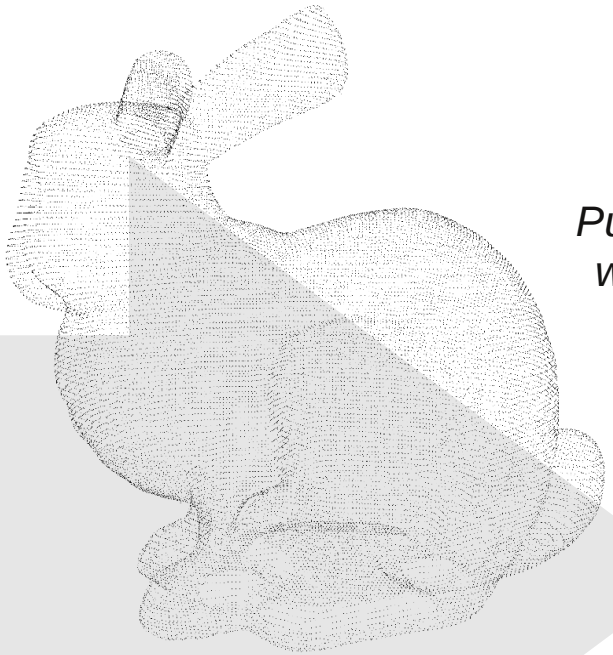
© C.-A. Bohn, FH Wedel, 2011

Wedel

© C.-A. Bohn, FH Wedel, 2011

„Objektrekonstruktion“ ?

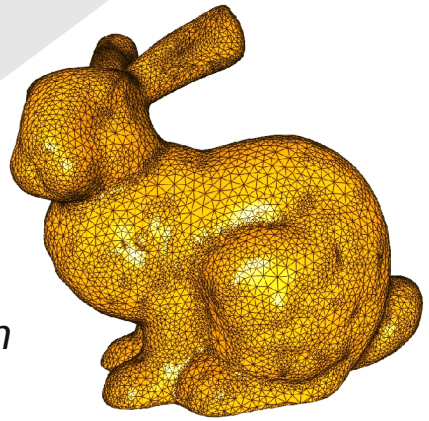
Laser-scanner



Punkte-wolke

Reales Objekt

Virtuelles Kaninchen

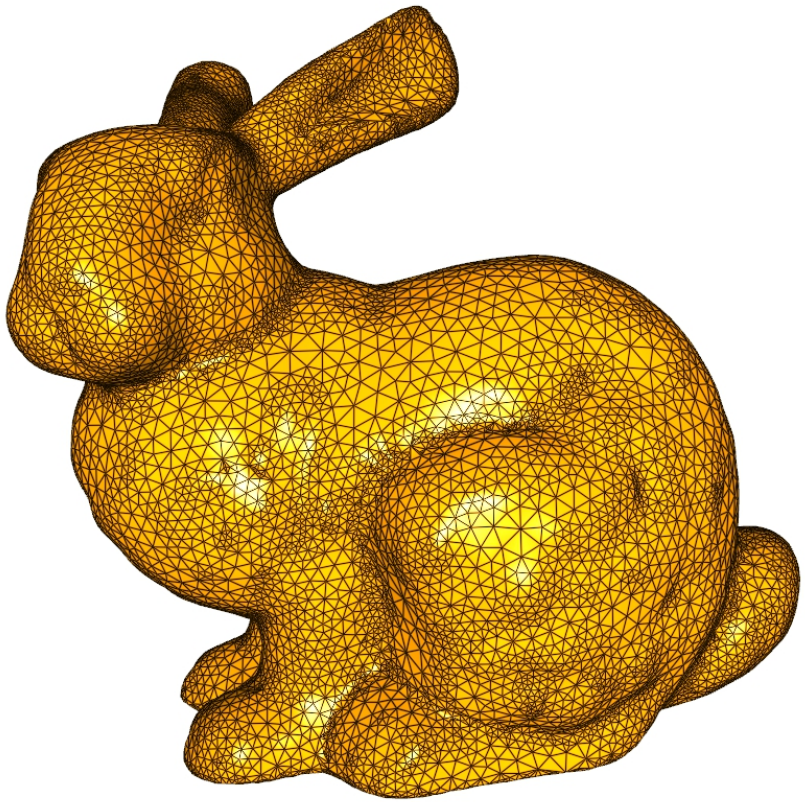


© C.-A. Bohn, FH Wedel, 2011

Objektrekonstruktion

In:
Gescannte
3D-Oberflächenpunkte

Out:
Polygonalisierung der
Objektoberfläche



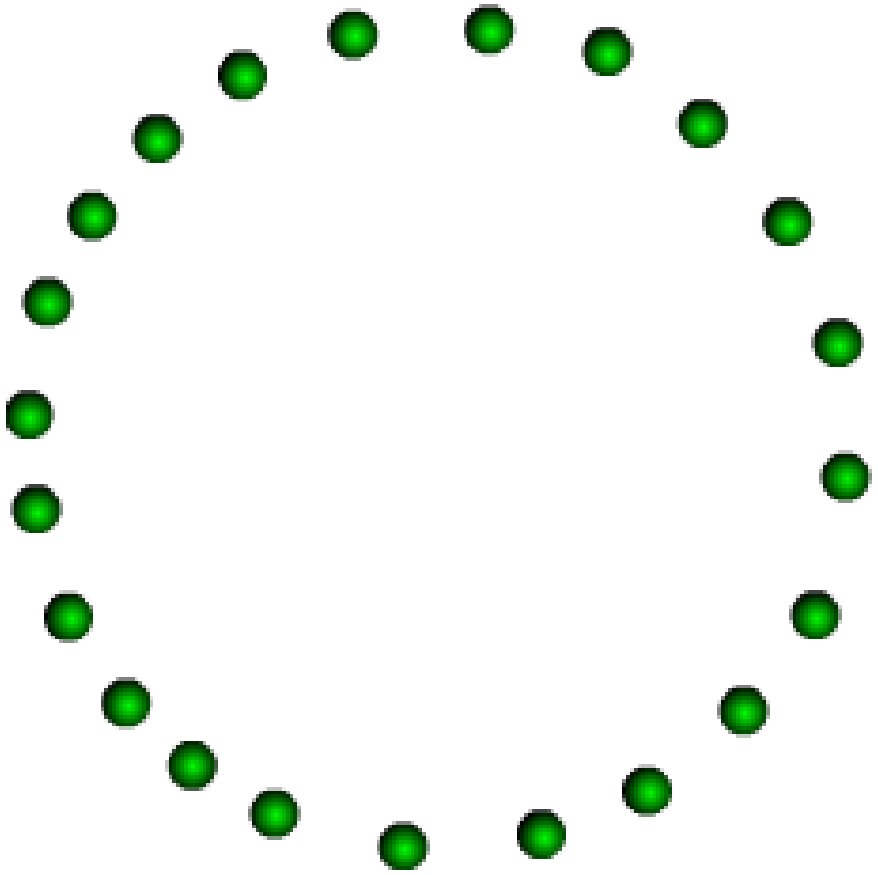
© C.-A. Bohn, FH Wedel, 2011

Anwendung der Objektregistrierung per Laser

- Vermessung von Gebäuden und Land
- Digitale Modelle für Film und Computerspiele
- Qualitätskontrolle bei der Produktion
- Konservierung
- Militärischer Bereich

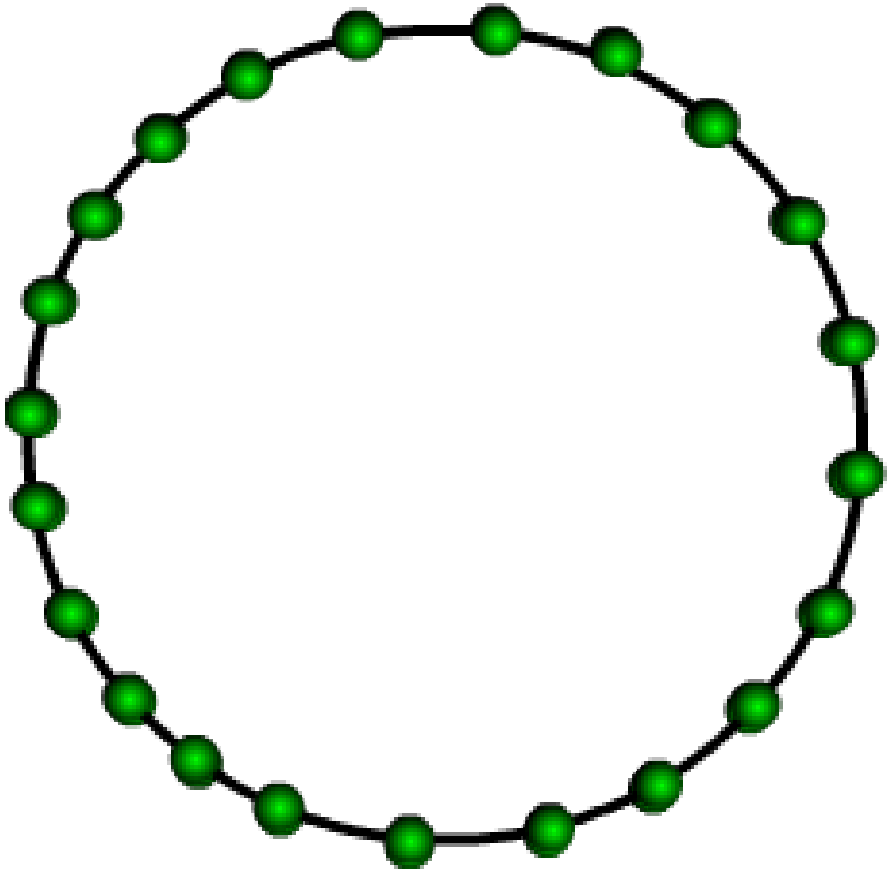
© C.-A. Bohn, FH Wedel, 2011

Oberfläche?



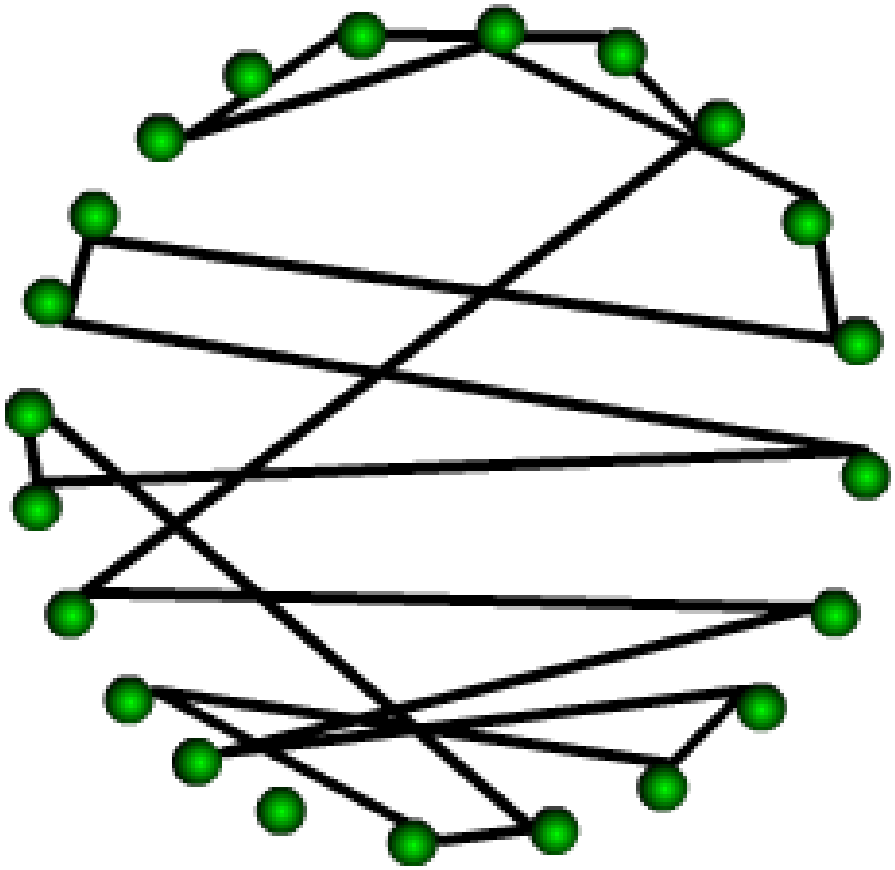
© C.-A. Bohn, FH Wedel, 2011

Die?



© C.-A. Bohn, FH Wedel, 2011

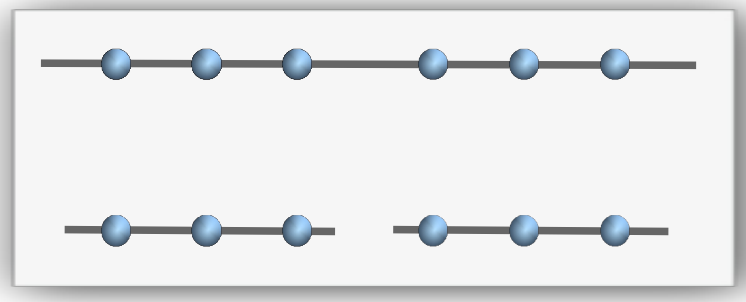
Oder die?



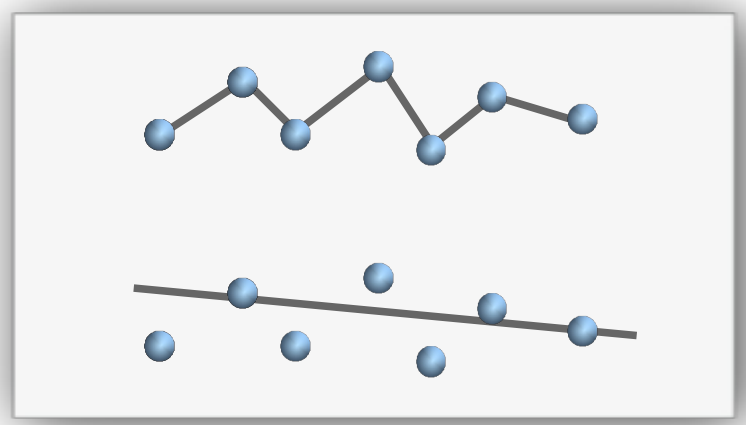
© C.-A. Bohn, FH Wedel, 2011

Von 1D nach 2D eindeutig?

- Loch oder nur großer Abstand der Samples?



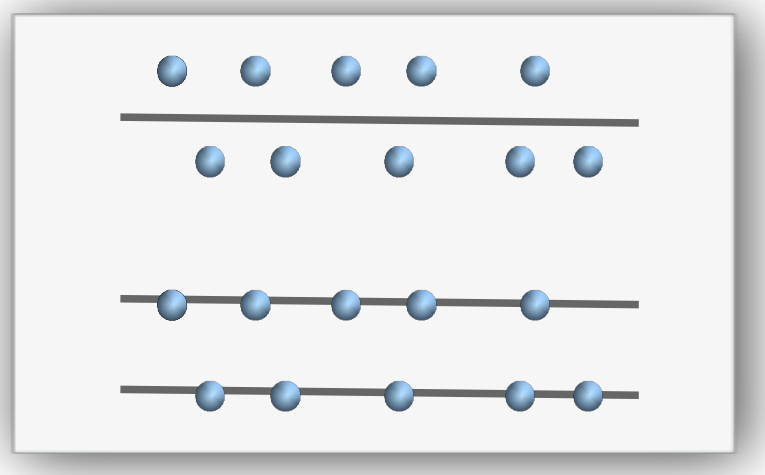
- Ab wann ist eine „Abweichung“ kein Rauschen mehr?



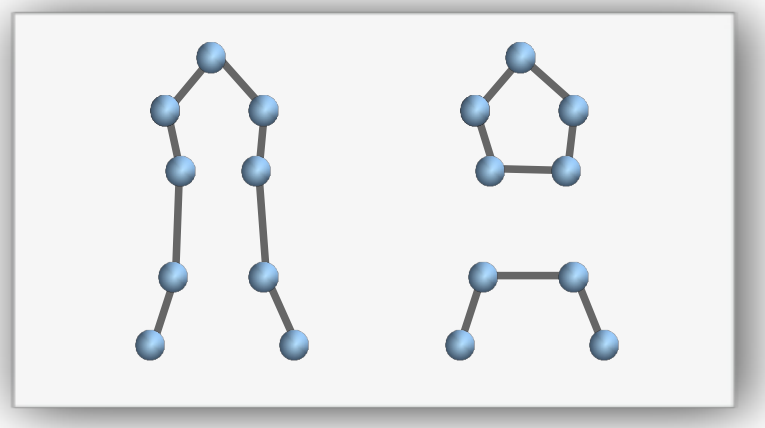
© C.-A. Bohn, FH Wedel, 2011

Von 1D nach 2D eindeutig?

- Dicke Objekte und Rauschen oder dünne Objekte?



- Kleine Formen oder separate kleine Objekte?



© C.-A. Bohn, FH Wedel, 2011

Definition eines Algorithmus': A) Eindeutigkeit

- *Problem:*
  - Punkte definieren eine Oberfläche nie eindeutig
  - Löcher, z.B. durch Verdeckung
  - Ungenauigkeiten, Rauschen bzw. Samples sind nicht exakt Punkte der gescannten Oberfläche
  - Unterabtastung (Punktauflösung zur grob, um eine feine Oberflächenstruktur wiederzugeben, z.B. Kanten, Ecken, Löcher)
  - Ungleichmäßige Verteilung, z.B. durch Mehrfach-Scan von unterschiedlichen Positionen aus
- *Folge*: Topologie der Oberfläche wird oft nicht korrekt erkannt
- *Abhilfe*: Benötigt werden lokale Regeln, die bestimmte Annahmen über die zu Grunde liegende Flächenstruktur aufgrund der Sample-Struktur machen

© C.-A. Bohn, FH Wedel, 2011

Definition eines Algorithmus': B) Datenstruktur

- Welche Datenstruktur? Dreiecke, Vierecke, parametrische Oberflächen höheren Grades, etc.?
- Anpassung der Granularität an Oberflächenstruktur möglich?
- Qualität bzw. sind Elemente „wohlgeformt“?
- Robustheit: Kann *jede* Oberfläche korrekt dargestellt werden?
- Schnelligkeit: Echtzeit, Interaktiv, Off-line oder langsam?

© C.-A. Bohn, FH Wedel, 2011

A+B = Smart Growing Cells

(A)

Lässt Spezifikation von Regeln
einfach zu

(B)

„Ideal“aufgrund Training eine
Künstlichen Neuronalen Netzes

© C.-A. Bohn, FH Wedel, 2011

Frühere Ansätze: 1) Volumen-basiert

- Prinzip
  1) Diskretisiere Raum in Volumenelemente (*Voxel*)
  2) Extrahiere Voxel, die Oberflächensamples enthalten
  3) Trianguliere Voxel

- Verfahren (Auswahl)
  - **Algorri & Schmidt, 1996**: zu 3) Voxel werden trianguliert und low-pass geflitert
  - **Hoppe et al., 1992**: Oberflächensamples erhalten distance-Wert (implizite Funktion), Triangulierung mit Marching Cubes
  - **Bajaj, Bernadini et al., 1997**: Startet mit einem Tetraeder, in dem Surface mittels distance-Wert der Samples als Polynom repräsentiert wird. Ist Approximation nicht ausreichend, wird Tetraeder in der Mitte unterteilt. Irreguläres Grid.
  - **Edelsbrunner, Mücke, 1994**: komplette Delaunay Tetrahedrization der Samples, dann löschen der großen mittels α-shapes

© C.-A. Bohn, FH Wedel, 2011

Frühere Ansätze: 2) Oberflächen-basiert

- Prinzip
  - Starte an einem lokalen Flächenstück
  - Taste Dich davon ausgehend über die Oberfläche
- Verfahren (Auswahl)
  - **Boissonnat, 1984**: Ein Sample und der naheste Nachhbar bilden die erste Kante. Punkt in Nachbarschaft, der mit der Kante den größten Winkel aufspannt bildet ein Dreieck.

© C.-A. Bohn, FH Wedel, 2011

Frühere Ansätze: 3) Warping

- Prinzip
  - Geometrie und Topologie eines „fertigen“ Polyeders wird an Sample-Wolke adaptiert
- Verfahren (Auswahl)
  - **Algorri, Schmitt, 1996**: Mesh ist Feder-Masse-Modell, Samples üben Gravitation aus
  - **Baader, Hirzinger, 1995**: Kohonen Feature Map adaptiert Sample-Menge

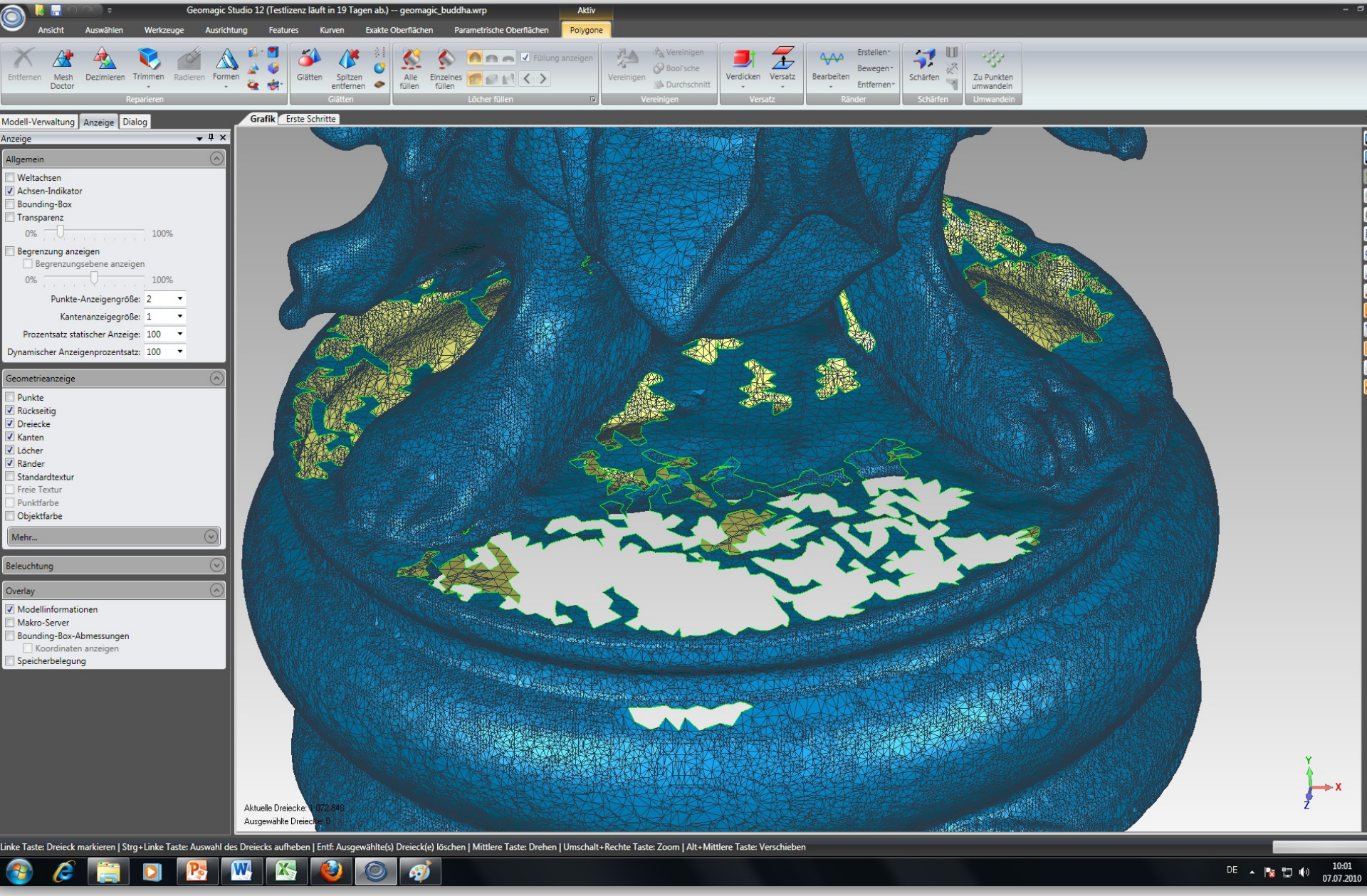
© C.-A. Bohn, FH Wedel, 2011

Probleme der meisten klassischen Verfahren

- Scheitern meist bei komplexen Homöomorphien oder separaten Körpern, da Bedingungen für eine Oberflächenstruktur nicht ausreichend spezifizierbar sind
- Keine inkrementellen Verhalten
  Probleme bei nicht-uniformen Sample-Verteilungen
- Probleme bei großen Sample Mengen

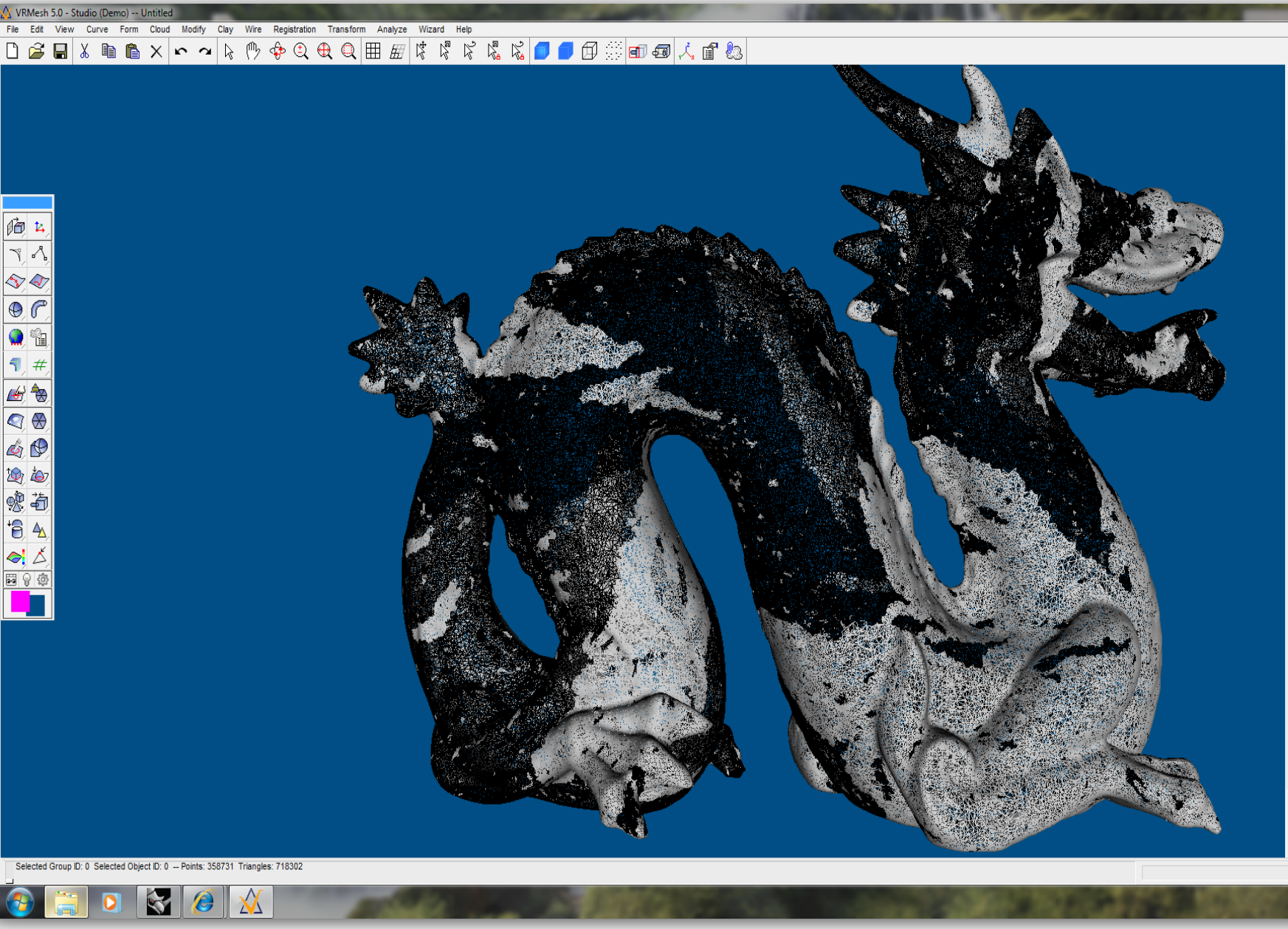
© C.-A. Bohn, FH Wedel, 2011

Kommerzielle Software



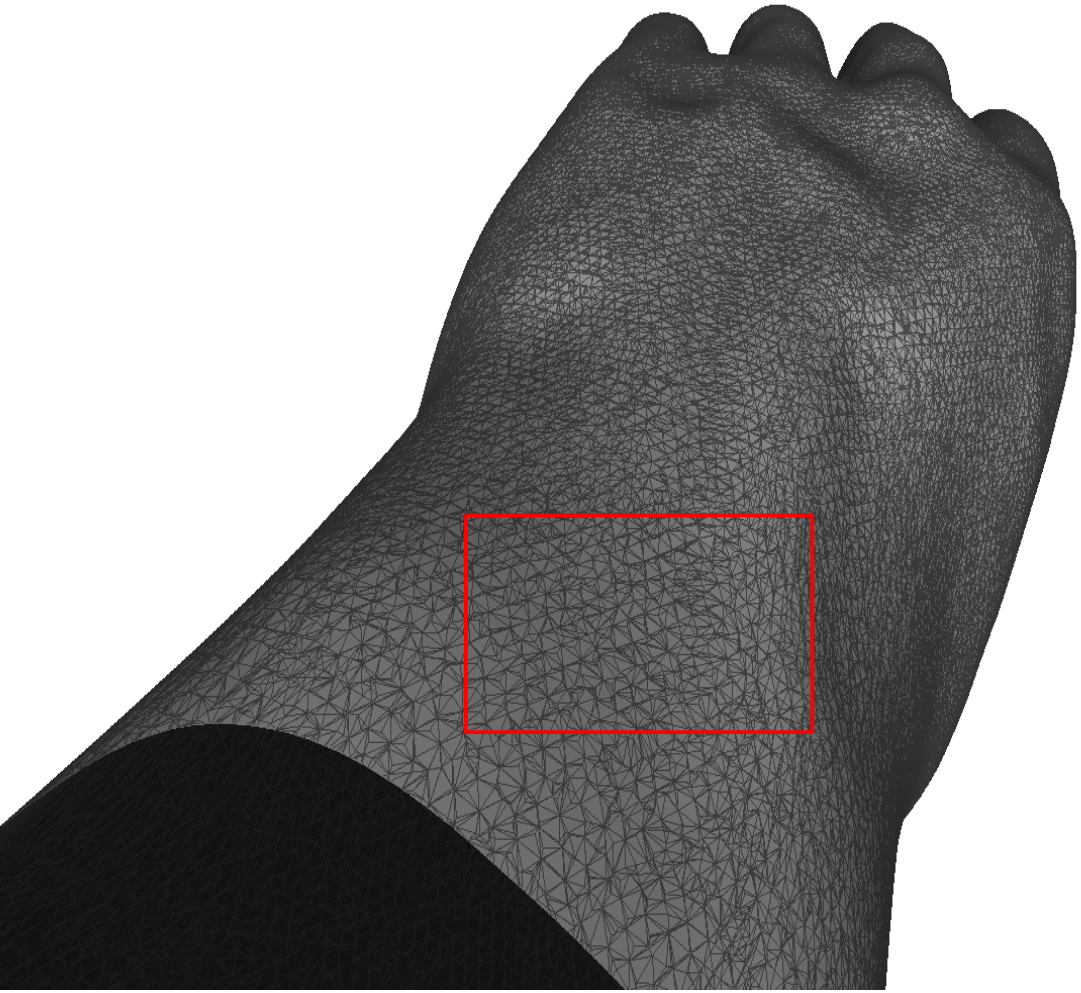
© C.-A. Bohn, FH Wedel, 2011

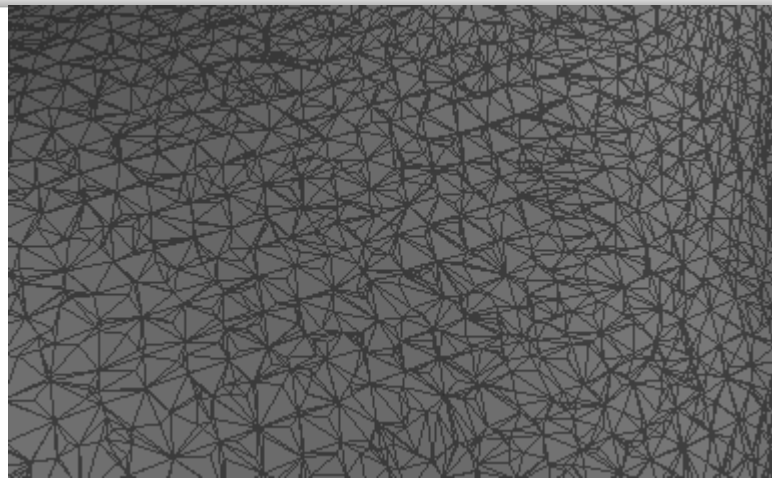
Kommerzielle Software



© C.-A. Bohn, FH Wedel, 2011

Kommerzielle Software





© C.-A. Bohn, FH Wedel, 2011

Smart Growing Cells (SGC)

- Definition von Regeln, die
  - nicht schwierig zu spezifizieren sind und die sich
  - in das Training einbetten lassen, ohne dessen Vorteile zu beeinflussen
- Vorteile der SGC für Rekonstruktion dadurch
  - Optimierung der Dreiecksqualität
  - Auflösung nicht eindeutiger Oberflächentopologien
  - Anpassung an Granularität der Oberflächenstruktur
  - Verarbeitung ungleichmäßiger Sample-Verteilungen
  - Plausibles Füllen von Löchern
  - Beliebige Topologien (z.B. auch separate Objekte)
  - Beliebige Sample-Mengen
  - Iterativ und inkrementell

© C.-A. Bohn, FH Wedel, 2011

„Gibt es keinen Algorithmus, nimm ein Neuronales Netz“

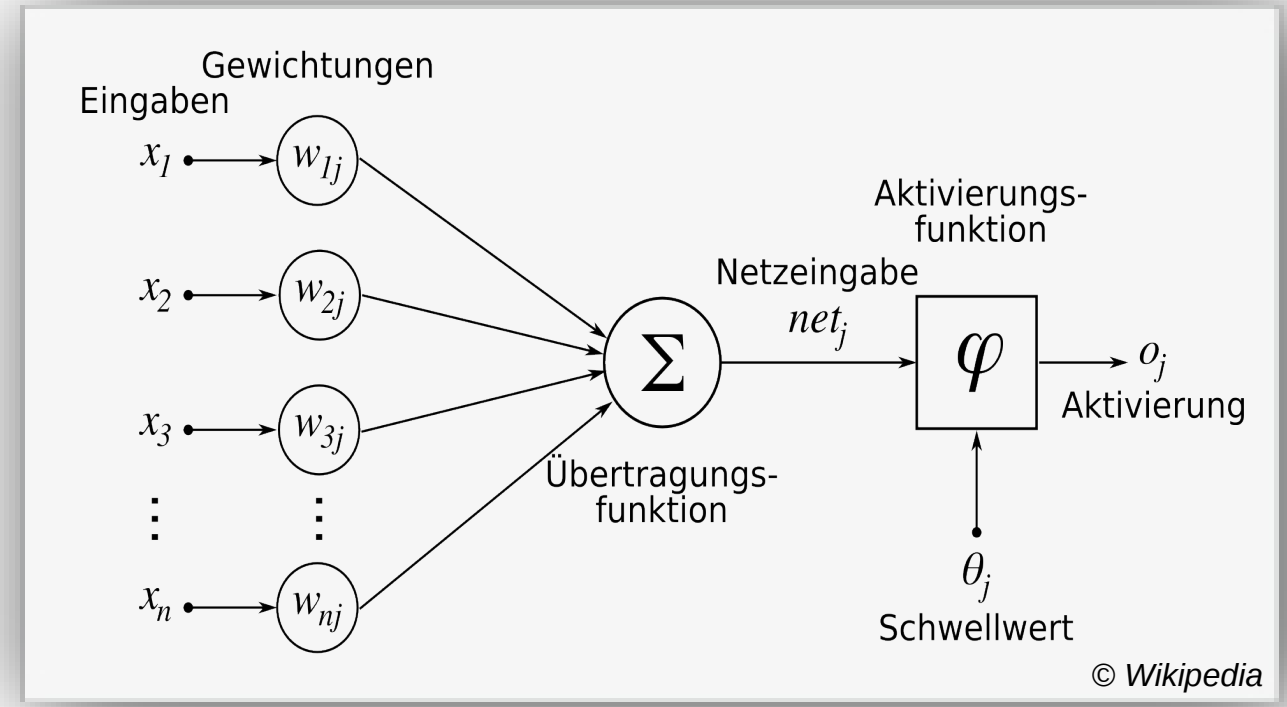
- „Simulieren das menschliche Gehirns“
- „Denken wie ein Mensch“
- „Lösen nach Bauchgefühl“

Tatsächlich:

- Approximieren *beliebige* Funktionen
- *Ein Algorithmus muss nicht spezifiziert werden*

© C.-A. Bohn, FH Wedel, 2011

Basis: ein künstliches Neuron



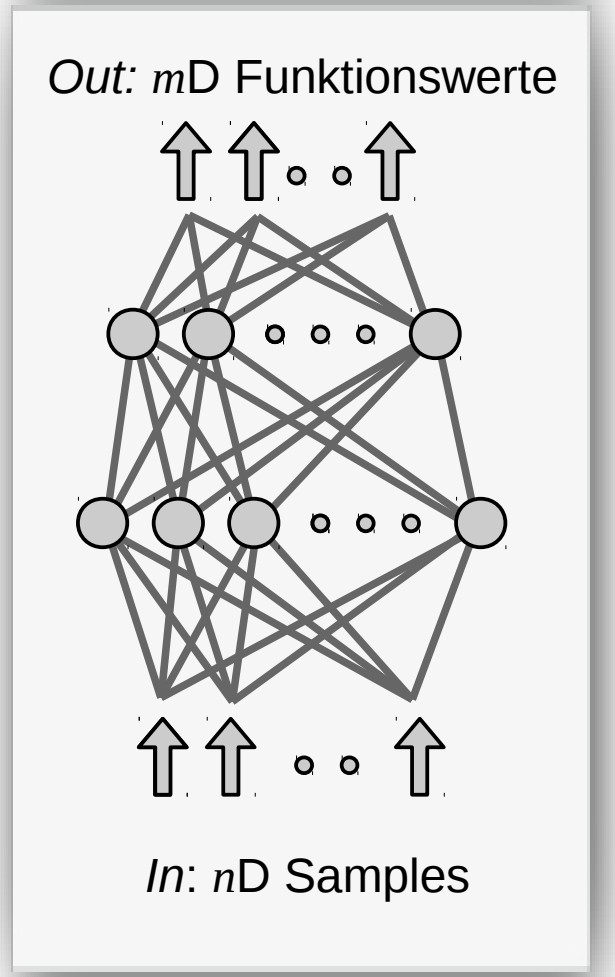
$$o_j = \varphi(net_j)$$

$$net_j = \sum_{i=1}^{n} w_{ij} x_i$$

© C.-A. Bohn, FH Wedel, 2011

Künstliches Neuronales Netz (KNN) für *Supervised Learning*

- Komplett vernetzte Layer von Neuronen
- Gewichte der Neuronenverbindungen werden durch In/Out-Paare trainiert
- Gewichte bestimmen schließlich die Funktion des Netzwerks
- Funktion: „Trainingsdaten werden gelernt, unbekannte Daten werden interpoliert“
- Eigenschaften
  - Sind in der Lage, komplizierte, nicht-lineare Zusammenhänge zu *lernen*, d.h. Diese müssen
  - nicht spezifiziert werden
  - *Auch*: Äquivalent zur Turing-Maschine



© C.-A. Bohn, FH Wedel, 2011

Supervised Training: Error-Backpropagation

- Input-Sample ξ liegt am Netz an => Output o_j an Ausgabeneuron j: $o_j = \varphi(net_j)$ mit $net_j = \sum_{i=1}^{n} w_{ij}$
- Fehler E im Vergleich zum gewünschten Output ζ

$$E = \sum_j (\zeta_j - o_j)^2$$

- Daraus Regel zum Adaptieren der Neuronengewichte

$$\Delta w_{ij} = -\eta \frac{\delta E}{\delta w_{ij}} = \eta \sigma_j x_j, \text{ mit}$$

$$\sigma_j = \begin{cases} \varphi'(net_j)(\zeta_i - o_j), & j \text{ Ausgabeneuron} \\ \varphi'(net_j) \sum_k \sigma_k w_{jk}, & \text{sonst} \end{cases}$$

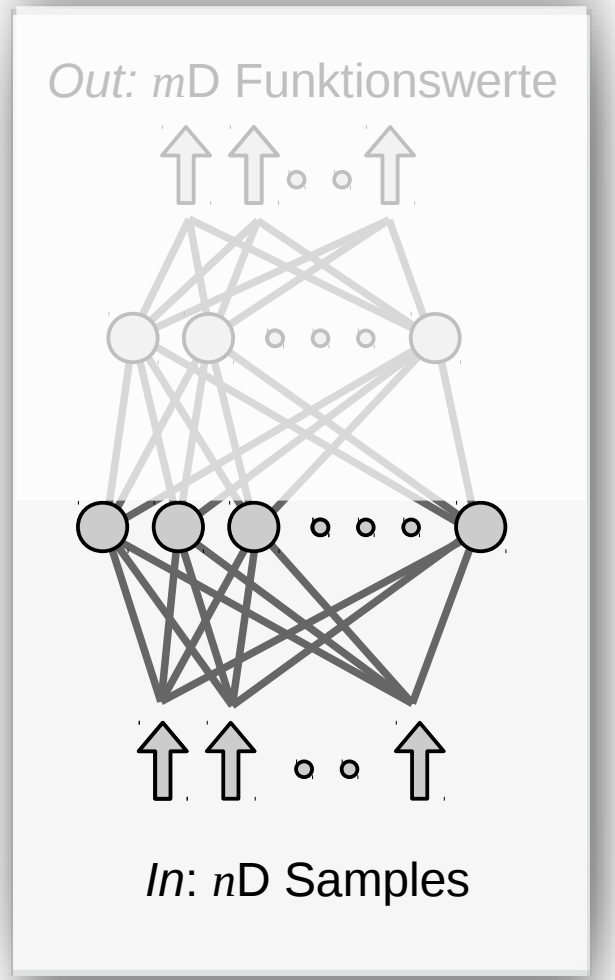
© C.-A. Bohn, FH Wedel, 2011

Unsupervised Learning (MacQueen 1967)

- Training:
  - Wähle zufällig Sample
  - *Ein* Neuron c_k spricht maximal darauf an
  - Trainiere das Netz, dass es diese Reaktion verstärkt

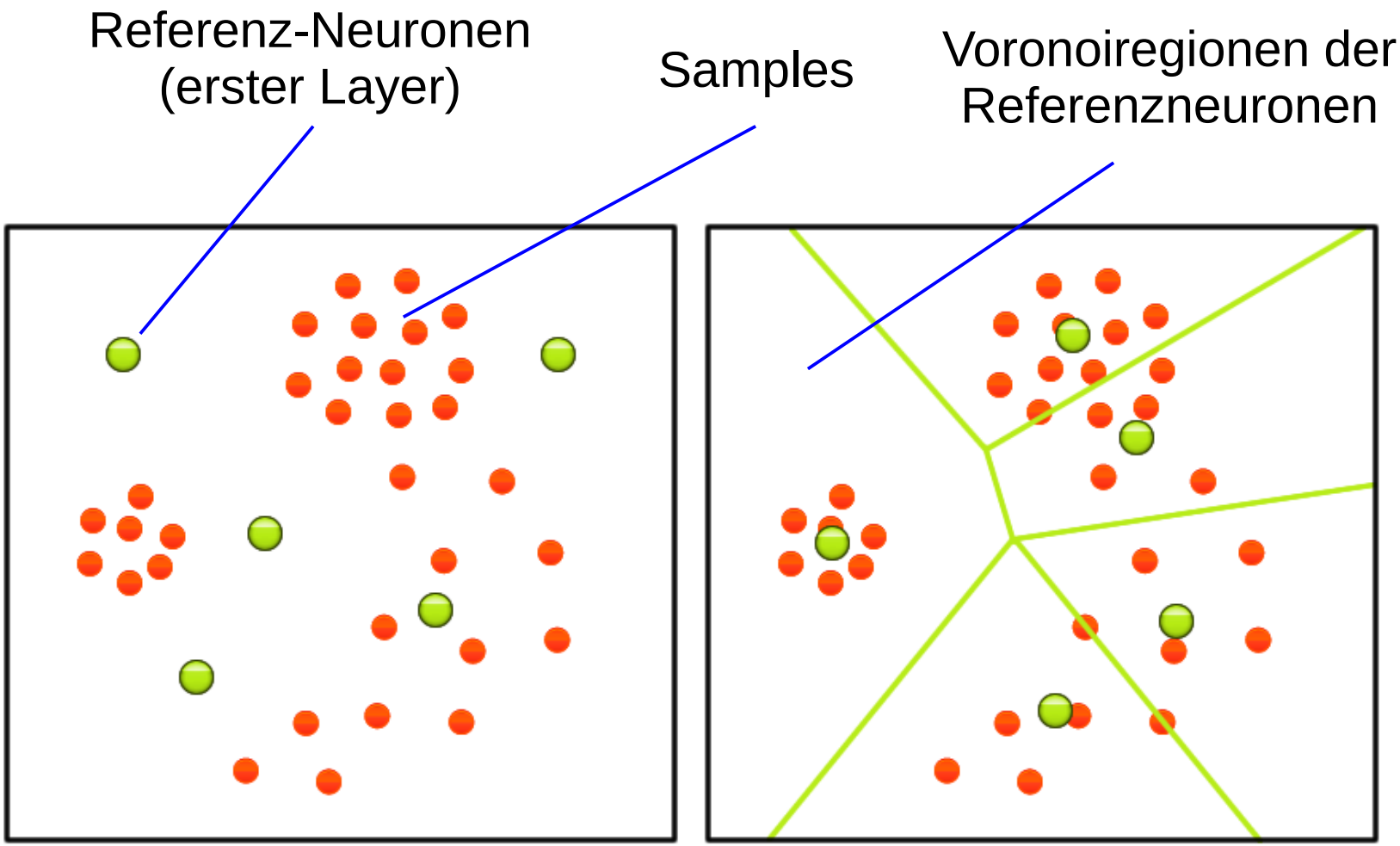
    bzw.

    bewege c_k in Richtung des Samples (gleiche Gewichte dem Sample an)
  - *Nach dem Training*: Klassifikation der Netzwerkfunktion per Hand
- Resultat
  - Bestimmte Sample-Gruppen aktivieren bestimmtes Neuron => Clustering, Dimensionalitätsreduktion
  - Allgemein: Sample-Verteilung wird durch Verteilung des Neuronen-Layers repräsentiert



© C.-A. Bohn, FH Wedel, 2011

Unsupervised Learning „von oben“



© C.-A. Bohn, FH Wedel, 2011

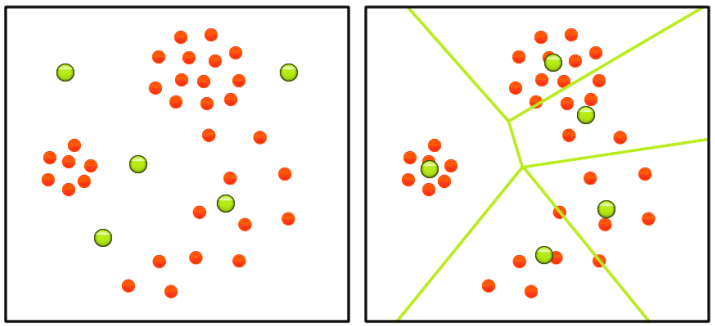
Unsupervised Learning

- *Gegeben:* Samples $s_i \in \mathbb{R}^n$

  Referenzneuronen $c_k \in \mathbb{R}^n, k = 0..m-1$

- Training:

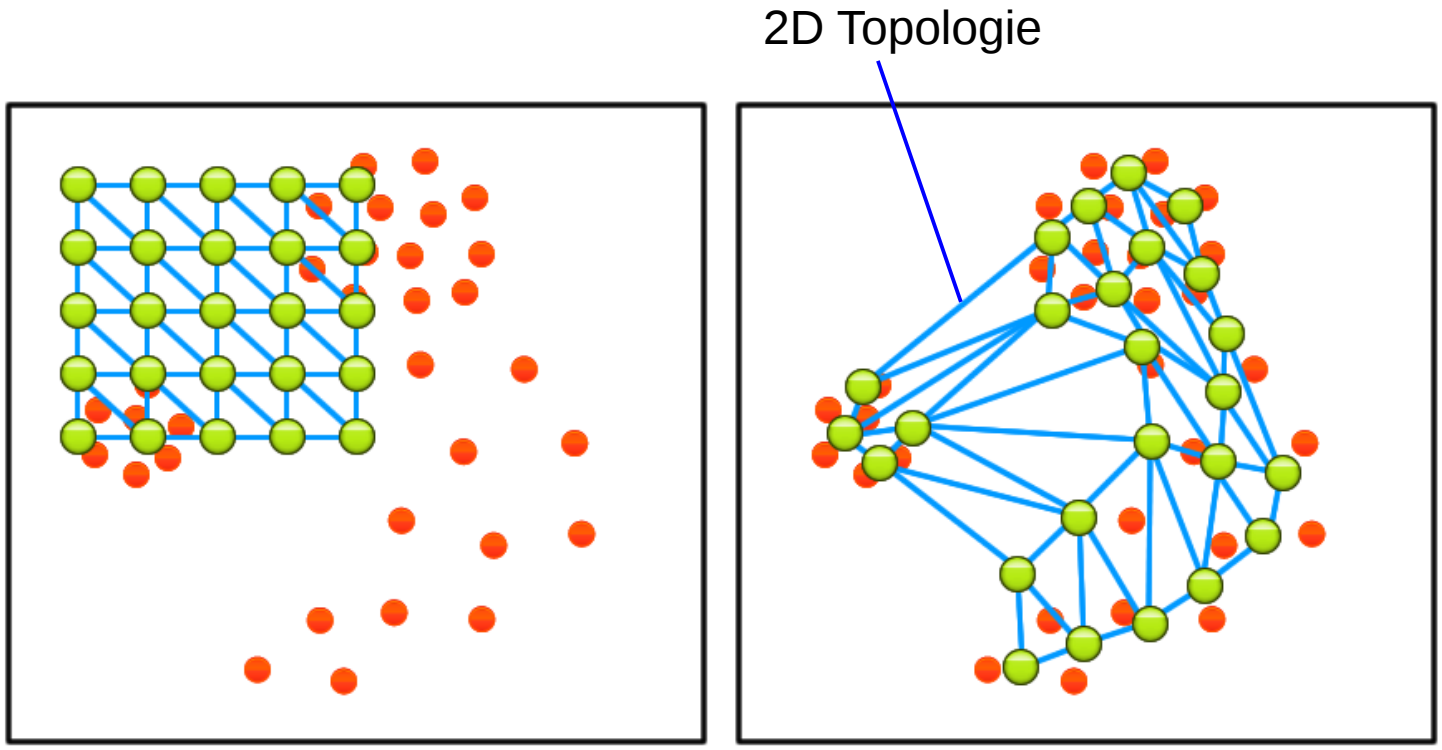
```
Loop:
   Wähle zufällig ein Sample s_i ∈ R^n
   Bestimme ähnlichste Referenzzelle c_bmu
   Korrigiere c_bmu in Richtung s_i  bzw. c_bmu = (1 − ε_bmu) c_bmu + ε_bmu s_i
   Reduziere ε_bmu
until: ε_bmu ist klein genug
```



© C.-A. Bohn, FH Wedel, 2011

Kohonens Self-Organizing Map (1982)

- Unsupervised Training, zusätzlich sind Neuronen in 2D Topology miteinander verknüpft … =>
- Topology wird mittrainiert



© C.-A. Bohn, FH Wedel, 2011

Kohonens Self-Organizing Map (1982)

- Training

  Loop:

  Wähle zufällig ein Sample $s_i \in \mathbb{R}^n$

  Bestimme ähnlichste Referenzzelle c_{bmu}

  Korrigiere c_{bmu} in Richtung s_i bzw. $c_{bmu} = (1 - \epsilon_{bmu}) c_{bmu} + \epsilon_{bmu} s_i$

  Loop über alle Nachbarn n_{bmu} von c_{bmu}

  Korrigiere n_{bmu} in Richtung s_i bzw. $n_{bmu} = (1 - \epsilon_n) n_{bmu} + \epsilon_n s_j$

  Reduziere ϵ_n

  Reduziere ϵ_{bmu}

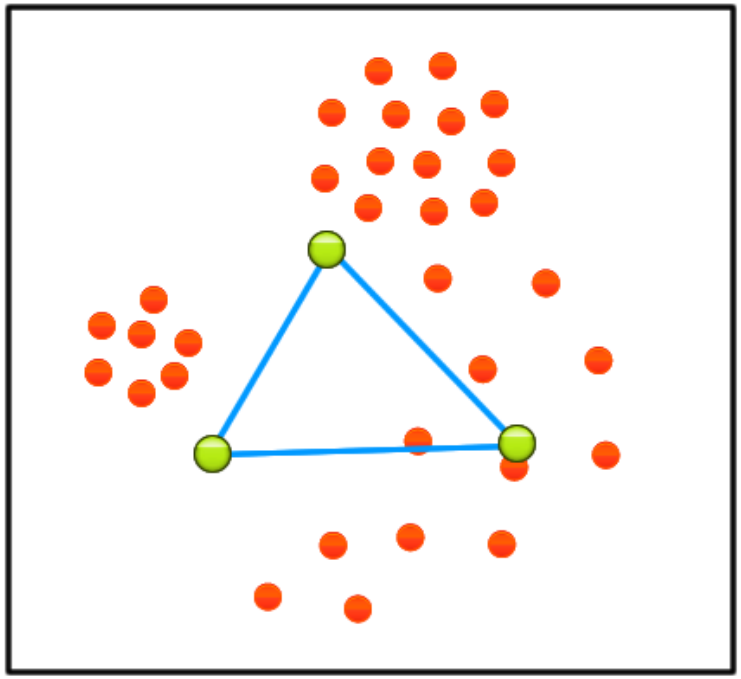
  until: ϵ_{bmu} ist klein genug

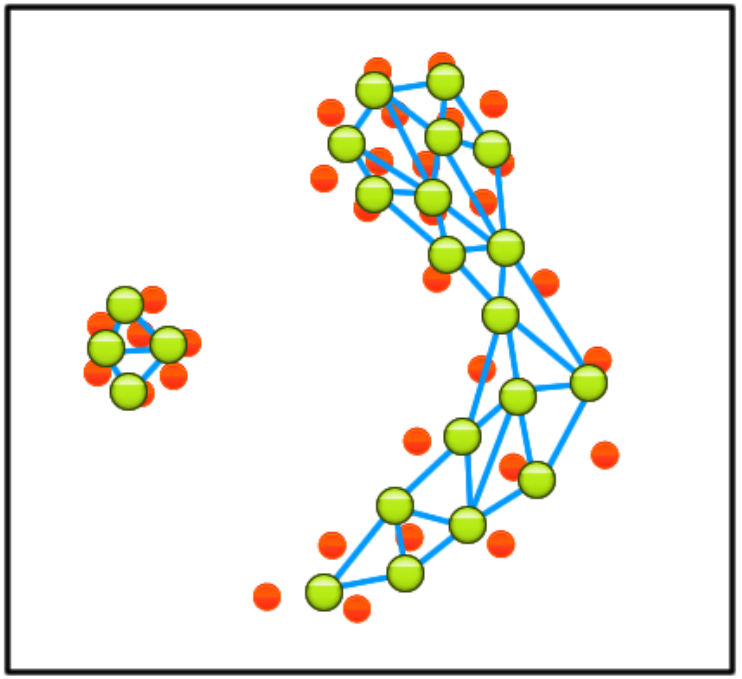
- Nachbarneuronen werden mitbewegt
- Implizite Topologie in Sample-Verteilung wird extrahiert

© C.-A. Bohn, FH Wedel, 2011

Fritzkes Growing Cells Structures Network (GCS, 1993)

- Topologie wächst und schrumpft nach Bedarf
- Dimensionalität ist beliebig (nicht nur 2D Map)





© C.-A. Bohn, FH Wedel, 2011

GCS: Training

Loop:

Wähle zufällig ein Sample $s_i \in \mathbb{R}^n$

Bestimme ähnlichste Referenzzelle c_{bmu}

Korrigiere c_{bmu} in Richtung s_i bzw. $c_{bmu} = (1 - \epsilon_{bmu}) c_{bmu} + \epsilon_{bmu} s_i$

Loop über alle Nachbarn n_{bmu} von c_{bmu}

Korrigiere n_{bmu} in Richtung s_i bzw. $n_{bmu} = (1 - \epsilon_n) n_{bmu} + \epsilon_n s_i$

Reduziere ϵ_n

Erhöhe Ressource-Term in c_{bmu}

Reduziere Ressource-Term in allen anderen Zellen

Ist Ressource-Term zu groß: füge Zelle ein

Ist Ressource-Term klein genug: lösche Zelle

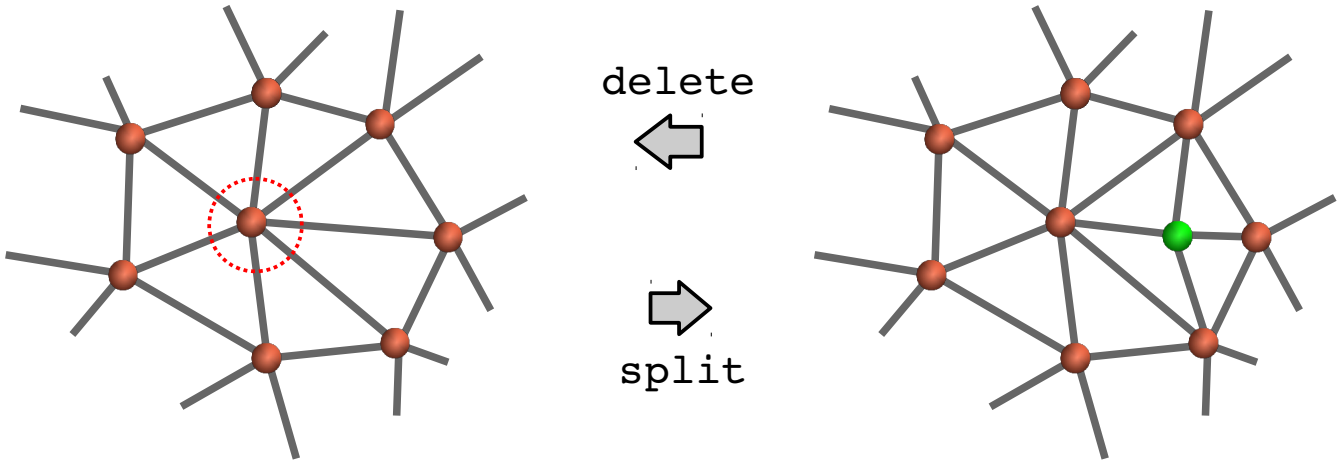
~~Reduziere ϵ_{bmu}~~

until: ϵ_{bmu} ist klein genug

© C.-A. Bohn, FH Wedel, 2011

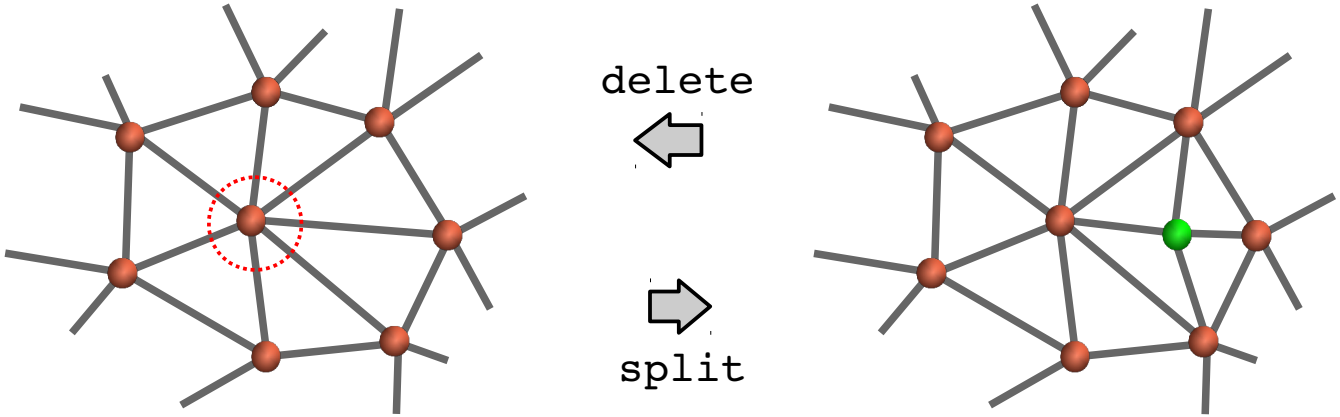
GCS: Ressource Term

- Ressource Term verfolgt die Häufigkeit, mit der eine Zelle selektiert wurde
- Er ist somit ein Hinweis auf die Sample-Verteilung
- Hohe Sample Verteilung => mehr Referenzzellen werden gebraucht, um Bereich besser zu repräsentieren => `vertex-split` Operation
- Niedriger Ressource Term => `vertex-delete` Operation



© C.-A. Bohn, FH Wedel, 2011

GCS: Vertex-Split & -Delete



split

1) Wähle Zelle mit höchsten Ressource-Term
2) Teile längste Kante und erzeuge neuen Vertex in der Mitte
3) Lösche alte Kanten und Flächen
4) Verbinde neuen Vertex mit allen beteiligten alten
5) Verteile alten Ressource-Term auf neue Zellen

delete

1) Wähle Zelle mit niedrigem Ressource-Term
2) Lösche Zelle, Kanten, Flächen
3) Verbinde alle beteiligten Zellen mit einem Vertex der gelöschten Kante
4) Erzeuge neue Kanten und Flächen

© C.-A. Bohn, FH Wedel, 2011

Fritzkes Growing Cells Structures Network (1990)

- Netzstruktur passt sich „Sample-Anforderungen“ an
- Topologie nun beliebig (nicht nur 2D)
- Startet mit Simplex in der Dimension der jeweiligen Toplogie (2D: Dreieck, 3D: Tetraeder)

- „Ressource“ kann auch „anders“ definiert sein
- Ressource wird Basis für Regeln für die Rekonstruktion

© C.-A. Bohn, FH Wedel, 2011

Smart Growing Cells (2010)

- *Basis*: Struktur eines SGC wird direkt als Polygonalisierung der zu erkennenden Objektoberfläche verwendet
- *Ziel*: Ressource-Term wird für zusätzliche Regeln, die z.B.über Lernverhalten, Löschen und Ergänzen von Referenzzellen entscheiden, umdefiniert aufgrund
  - Uneindeutigkeiten (Löcher, Kanten, Ecken, Objektgrenzen)
  - Besonderer Anforderung an die Oberfläche
- Historische Arbeiten
  - **Baader, Hirzinger, 1995**: Kohonen Feature Map für Rekonstrukion
  - **Bohn, 1996**: Growing Cell Structures für Rekonstruktion
  - **Ivrissimitzis, Seidel 2003**: Growing Cell Structures für Rekonstruktion

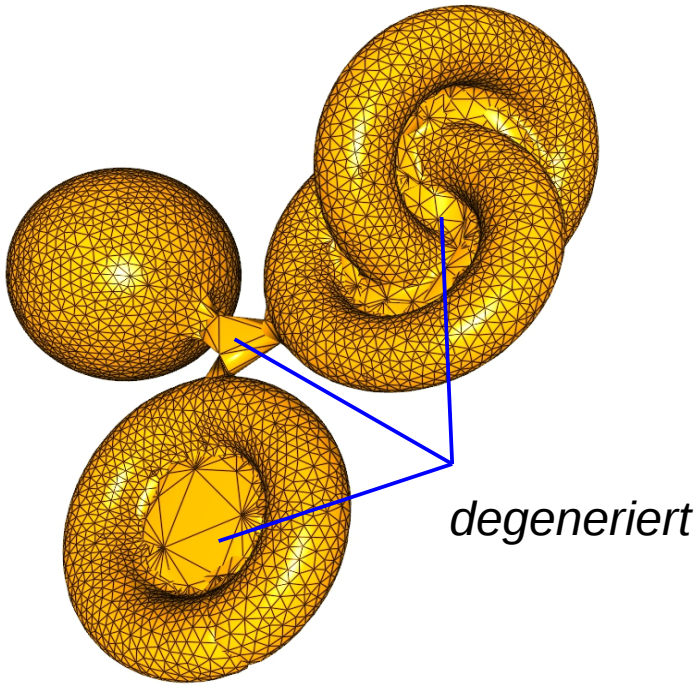
© C.-A. Bohn, FH Wedel, 2011

SGC: „Aggressive Cut“

- GCS können Zellen einfügen oder löschen, wenn Ressource-Term groß bzw. klein genug ist
- SGC löschen *auch dann*, wenn Bereiche des Netzes „ungünstig („*degeneriert*“) bzgl. der Netzstruktur“ sind =>

Training wird durch anwendungs-bezogene Regeln ergänzt

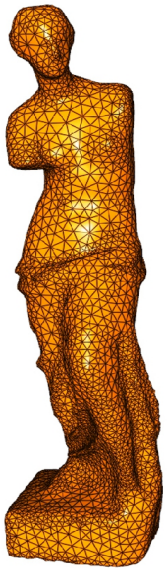
- SGC löschen „lieber zu viel als zu wenig“
- Degenerierte Regionen sind ein perfekter Indikator für inkonsistentes Netz
- *Hier*: Homöomorphiekorrektur

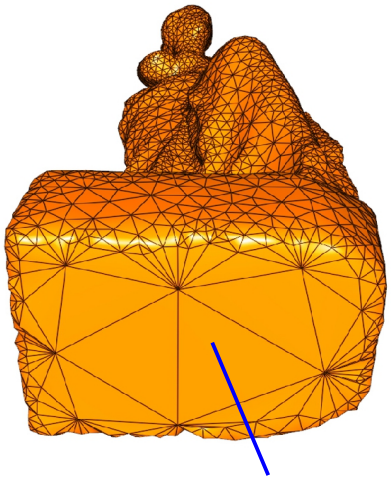


© C.-A. Bohn, FH Wedel, 2011

SGC: „Aggressive Cut“

- *Hier*:
  Detektion von Löchern
  in Sample-Verteilung
- Netzstruktur durch
  Rand-Sample über
  das Loch „gedrängt“
  => degeneriert





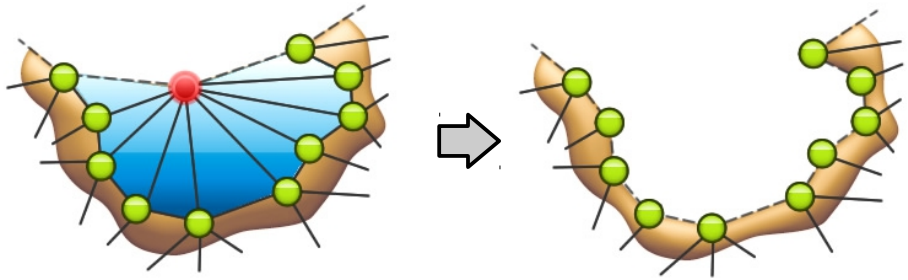
- Detektion:
  - *Bedingung 1*: Lokale Vertex-Valenz erreicht bestimmte Größe
  - *Bedingung 2*: Dreieck ist zu *uneinheitlich* (inkonsistent) (Verhältnis zwischen Flächen des Dreiecks und gleichseitigem Vergleichsdreieck aus längster Kante ist klein)
  - Degeneriert, wenn Bedingungen 1 und 2 für ein Dreieck und mindestens eines seiner Nachbarn gilt

© C.-A. Bohn, FH Wedel, 2011

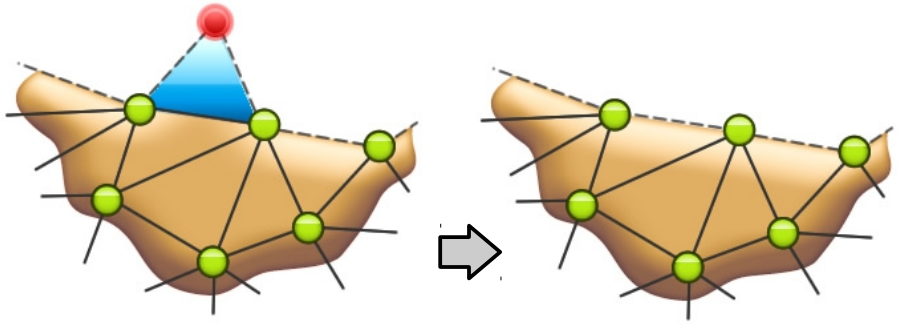
SGC: „Aggressive Cut“ & Heilung

- Aggressive Cut erzeut unschöne Netzstruktur, die geheilt werden muss

- *Nasty Vertex*
  - hohe Valenz und/oder
  - hohe Anzahl an lokalen inkonsistenten Dreiecken



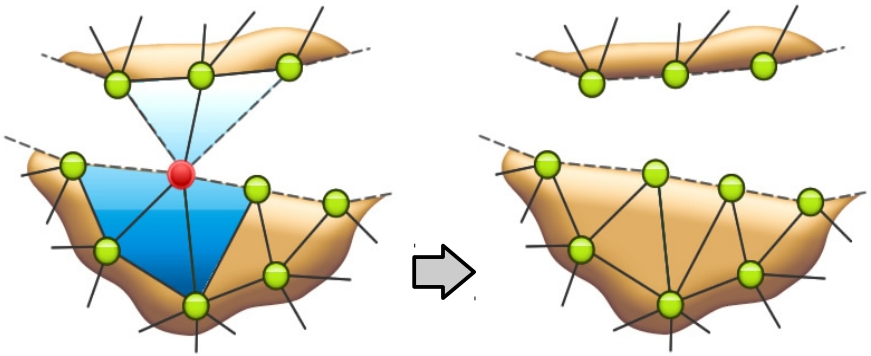
- *Spike*
  - Valenz ist 2



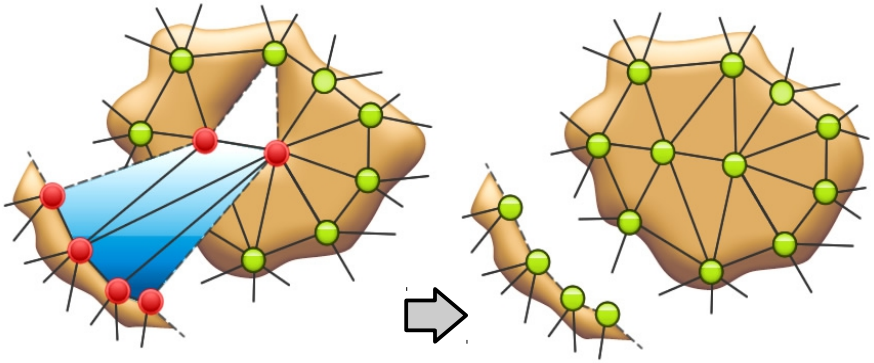
© C.-A. Bohn, FH Wedel, 2011

SGC: „Aggressive Cut“ & Heilung

- *Needle Eye*
  - Zelle, die mit zwei Netzrändern verbunden ist



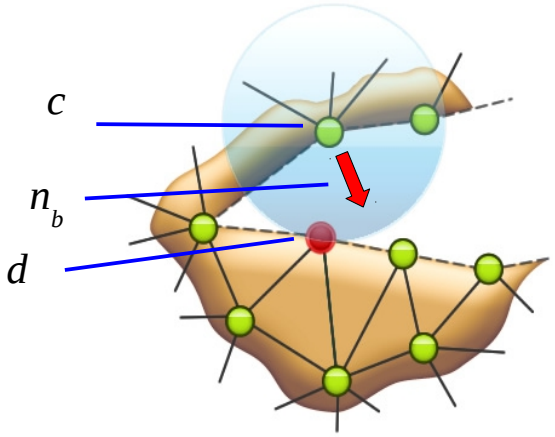
- *Bridge*
  - Kante mit mehr als zwei Dreiecken



© C.-A. Bohn, FH Wedel, 2011

SGC: Verschmelzung

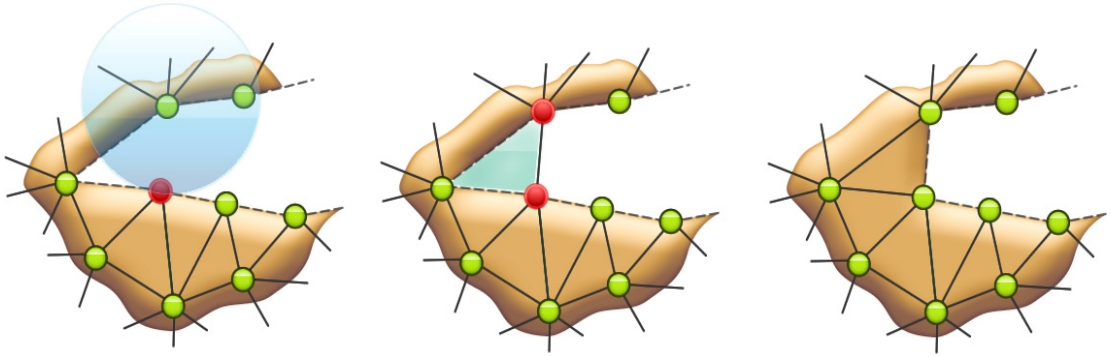
- „Gegenstück zu Aggressive Cut“
- Netz-Topologie liefert Kriterium, dass Zellen mit anderen verschmelzen können
- Für zu untersuchendes Zellpaar c und d am einem Rand
  - Definiere Suchkreis um c im Mittel zweier benachbarter Zellen
  - Definiere Radius r als Mittel der Abstände zu allen Nachbarn
  - Definiere „Randnormale“ n_{bc} aus dem Mittel aller Kanten zu allen Nachbarn
- c wird mit einer Zelle d verschmolzen, wenn
  - d im Suchkreis liegt
  - n_{bc} in die Richtung von d zeigt
  - n_{bd} in die Richtung von c zeigt



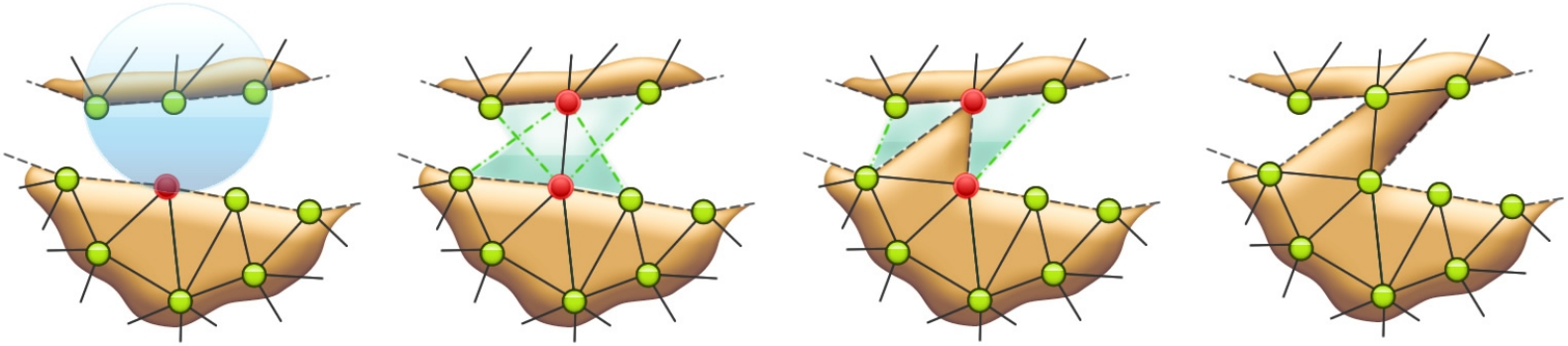
© C.-A. Bohn, FH Wedel, 2011

SGC: Verschmelzungsprozess

- c und d haben einen gemeinsamen Nachbarn



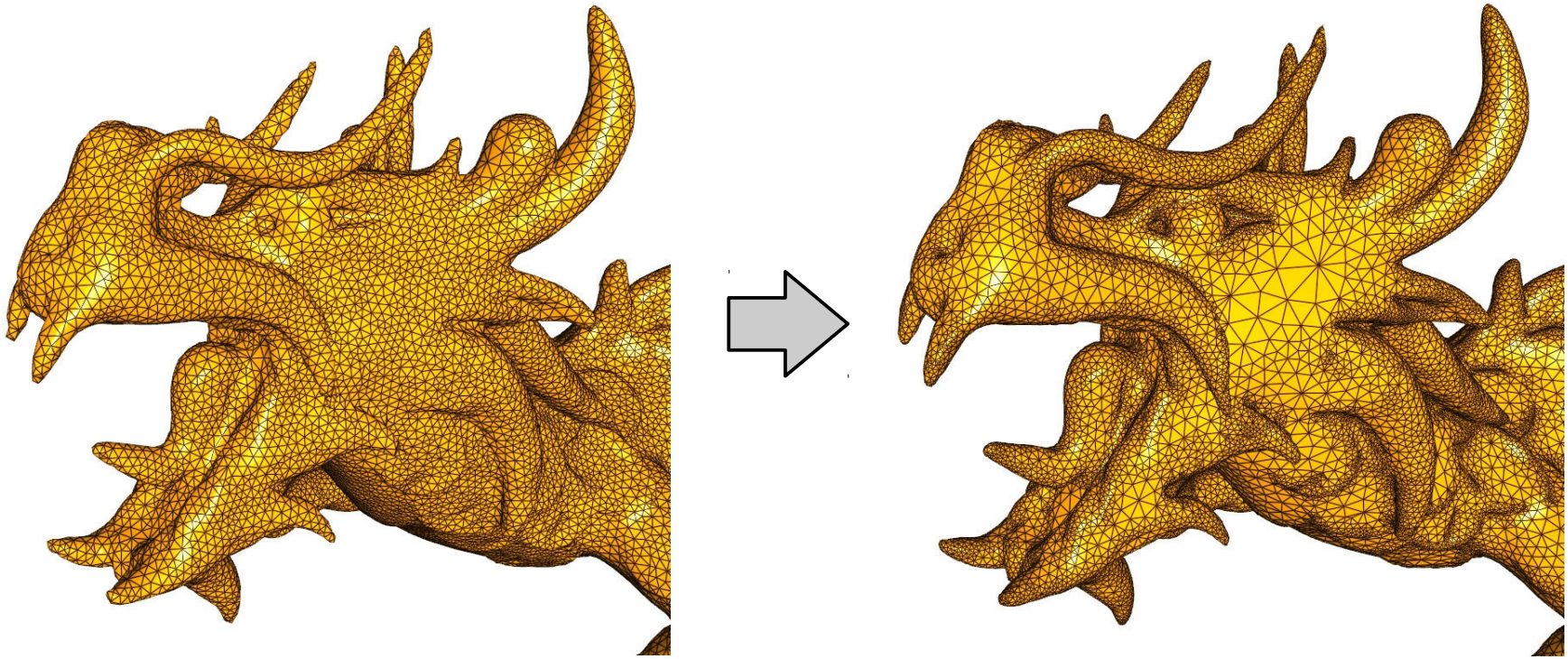
- c und d liegen auf getrennten Rändern => Zellen, die das einheitlichste Dreieck (Kanten variieren am wenigsten) bilden, bilden neue Dreiecke



© C.-A. Bohn, FH Wedel, 2011

SGC: Anpassung an Oberflächenkurvigkeit

- Ziel: Stärker variierende Oberflächenstrukturen sollen höhere lokale Netzgranularität erhalten



© C.-A. Bohn, FH Wedel, 2011

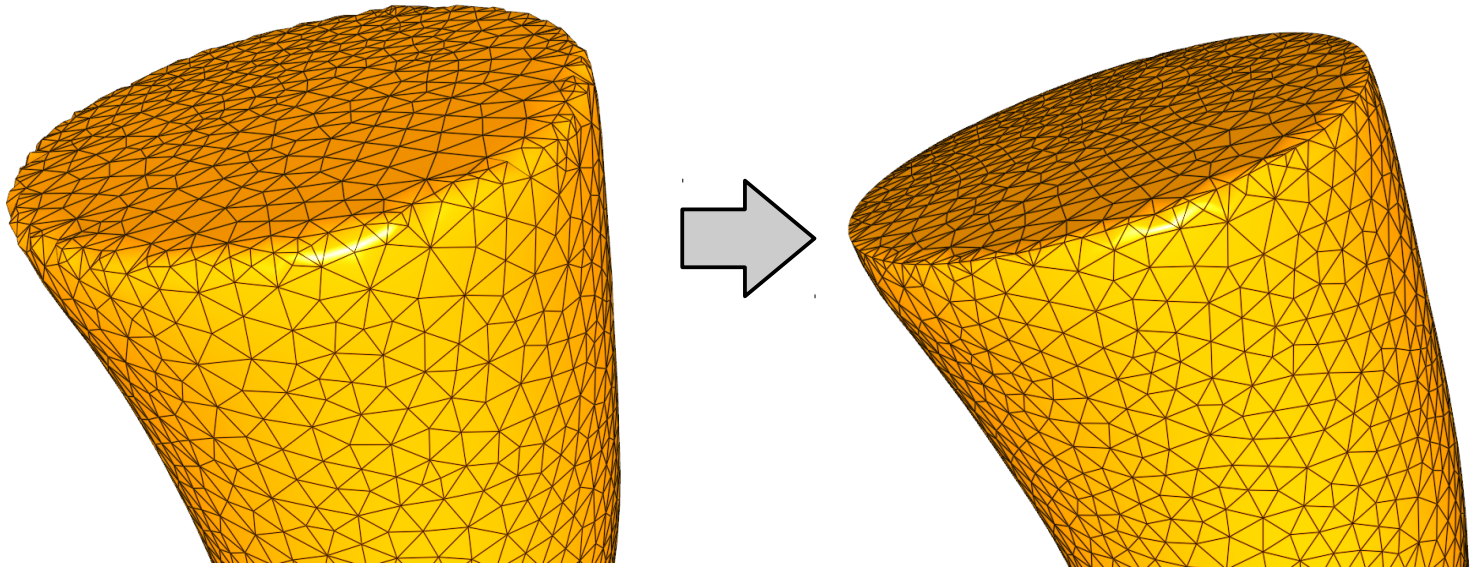
SGC: Anpassung an Oberflächenkurvigkeit

- Vorteile
  - Effizienteres Matching der Flächenstruktur mit der Netzgranularität
  - Größere Unabhängigkeit von Sample-Verteilung (Löcher, unregelmäßige Sample-Verteilungen, etc.)
- Algorithmus:
  - Bilde S_n Mittelwert der Skalarprodukte der Normalen einer Zelle mit denen ihrer Nachbarn => „*Curvature*“-Parameter
  - Erhöhe Ressource-Term um 1 - S_n
  - Kleiner Ressource-Term führt zur Reduzierung der Zahl der Zellen an dieser Stelle
  - Unterer Schwellwert für Ressource-Term sorgt dafür, dass ebene Flächen nicht komplett gelöscht werden

© C.-A. Bohn, FH Wedel, 2011

SGC: Discontinuity-Cells

- Kanten und Ecken sind schwierig zu erkennen und zu modellieren, da Zellen eher Mittelwerte repräsentieren und Kanten mehr Samples benötigten, um adäquat repräsentiert zu werden



© C.-A. Bohn, FH Wedel, 2011

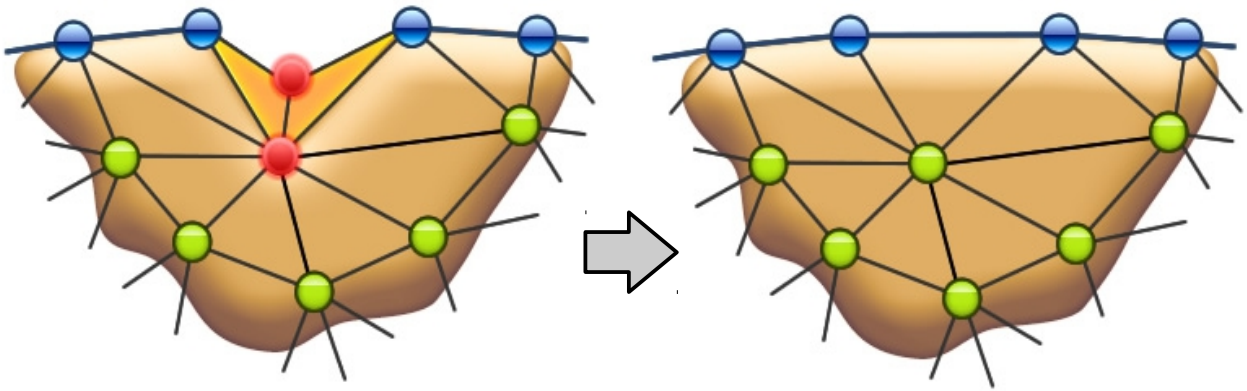
SGC: Discontinuity-Cells

- Kantendetektion:
  - Finde Zelle c mit hoher Curvature
  - Vergleiche mittleren Gradienten des zweiten Rings um c mit Gradienten bei c
  - Ist der Unterschied groß („Knick“ zwischen Flächen wird vermutet), so ändert sich die Zelle zu einer „Discontinuity Cell“ mit speziellen Lernregeln
- Kantenmodellierung
  - Discontinuity Cells werden nur noch in Richtung in Ihrer Normalen bewegt

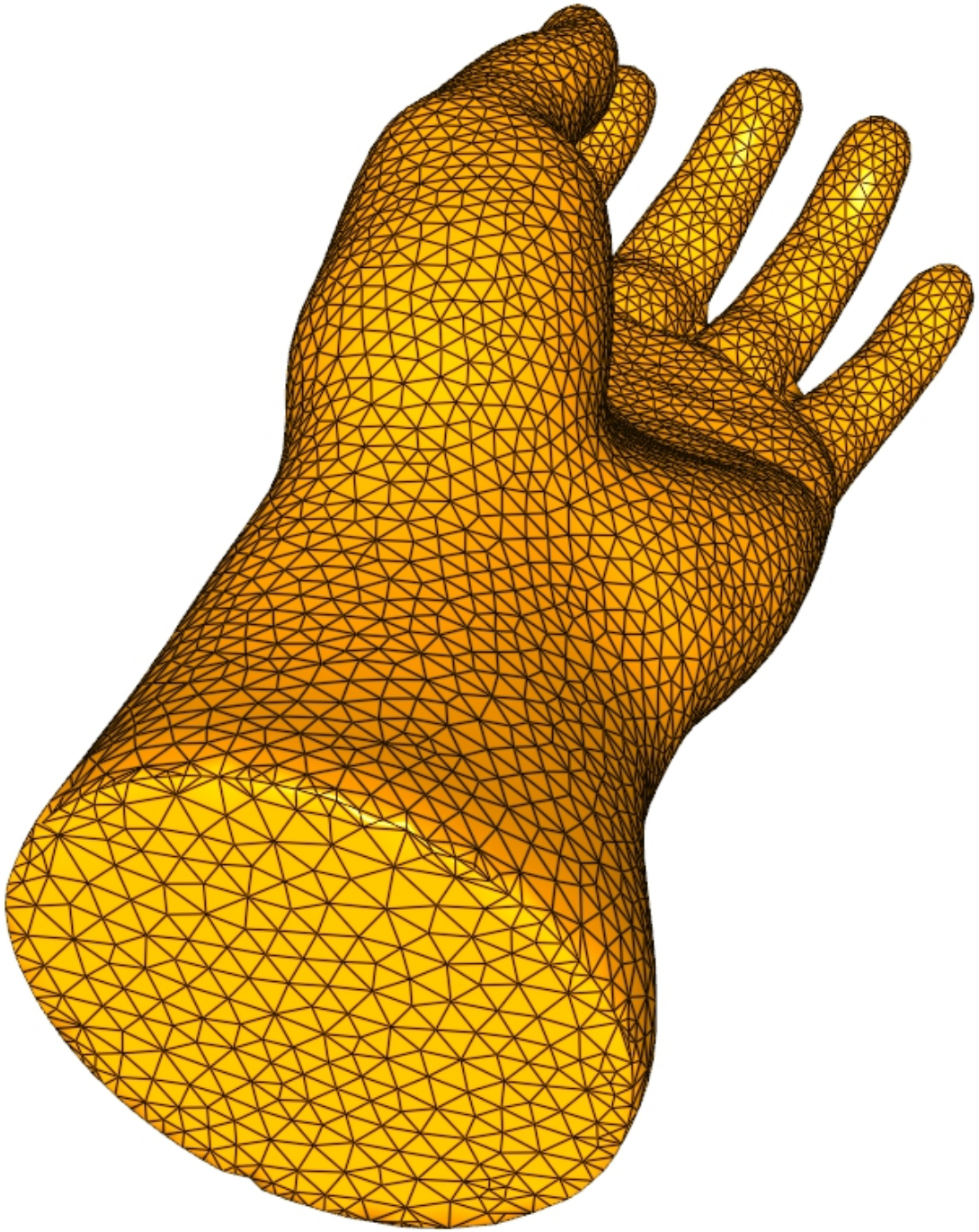
© C.-A. Bohn, FH Wedel, 2011

SGC: Discontinuity-Cells

- Nicht entdeckte Zellen werden entdeckt durch die Varianz stark unterschiedliche Dreiecksnormalen dreier aufeinanderfolgender Discontinuity Cells
- Edge Swap testet, ob diese vermindert werden kann

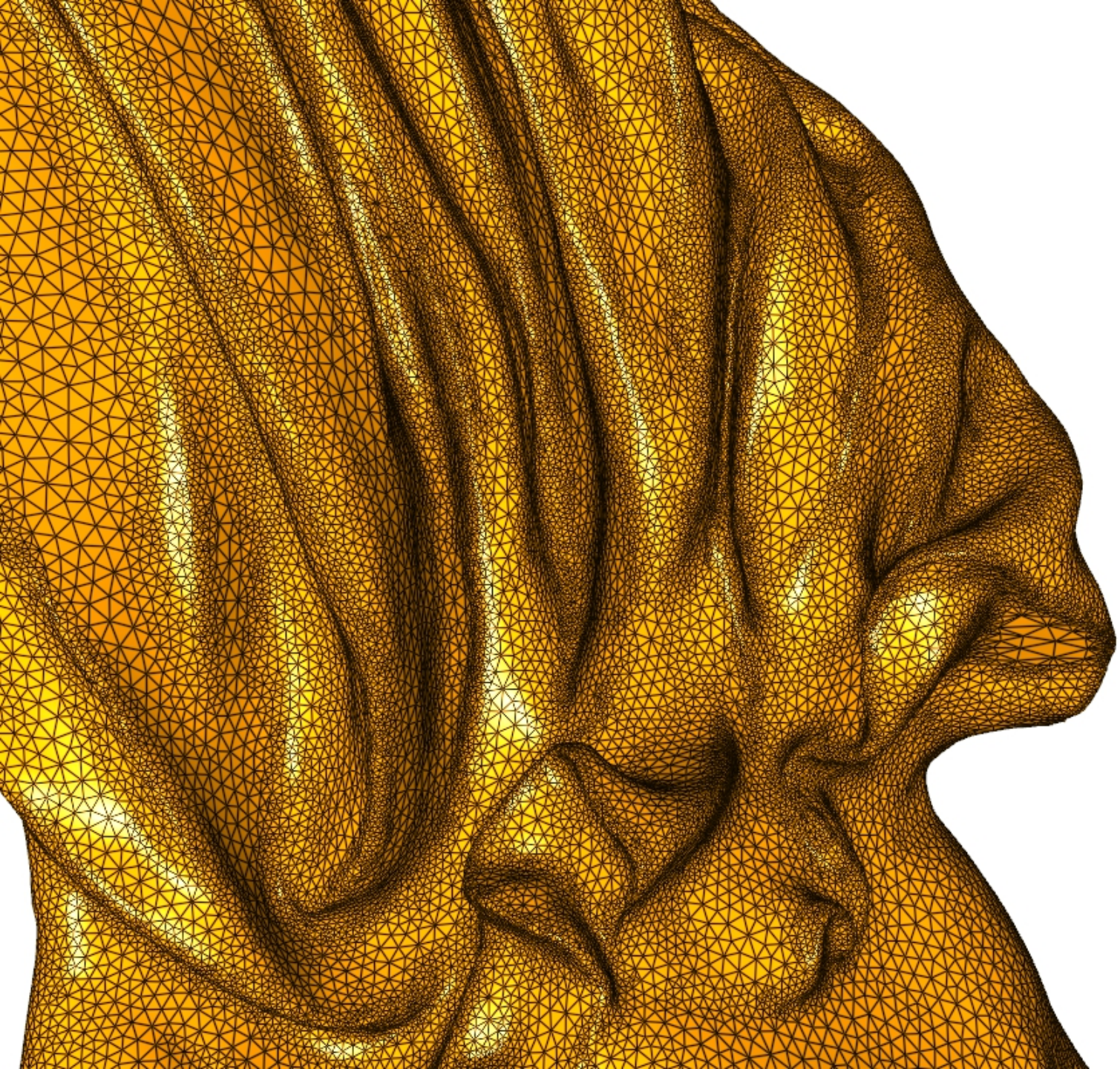


© C.-A. Bohn, FH Wedel, 2011

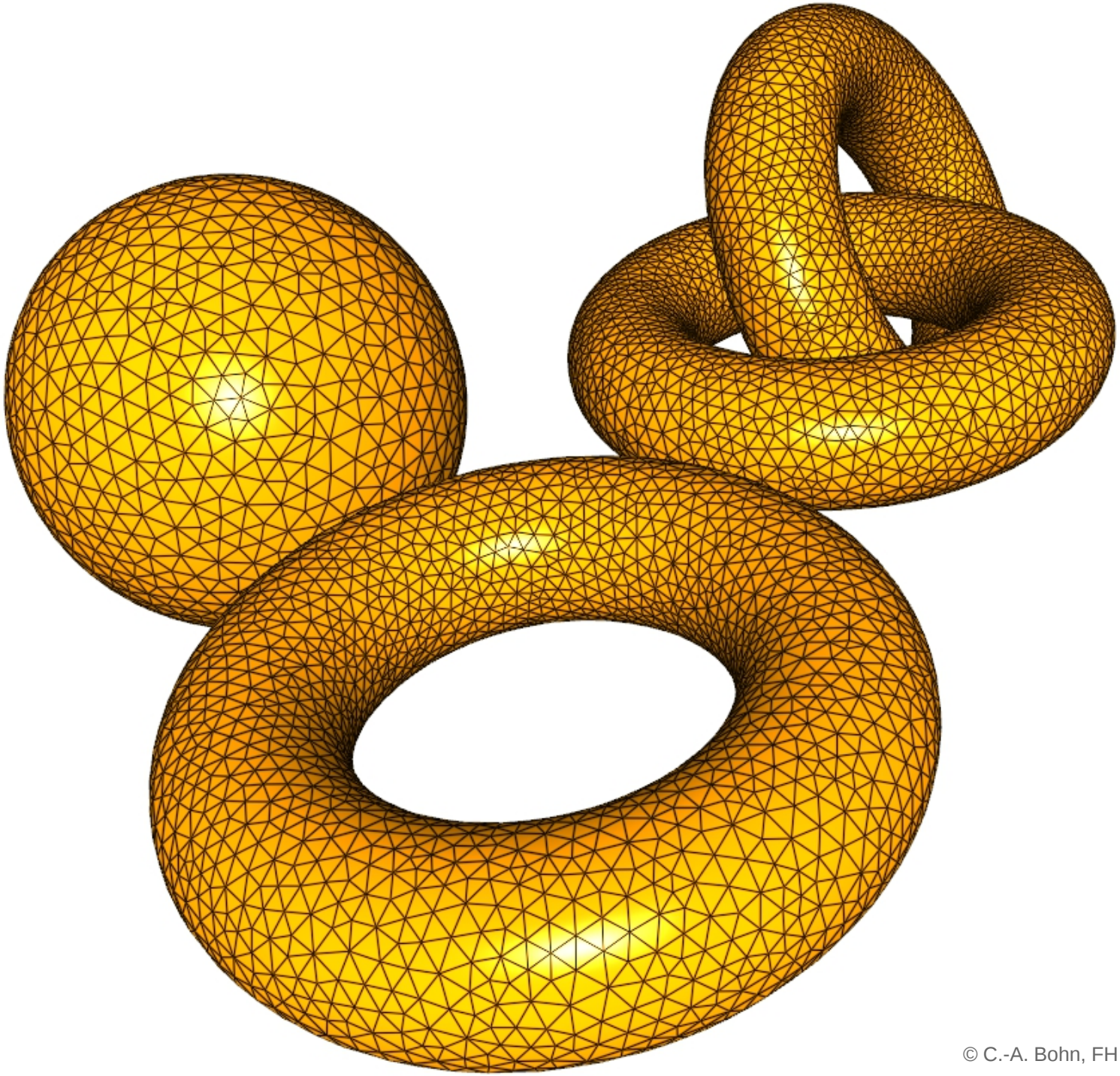


© C.-A. Bohn, FH Wedel, 2011

© C.-A. Bohn, FH Wedel, 2011

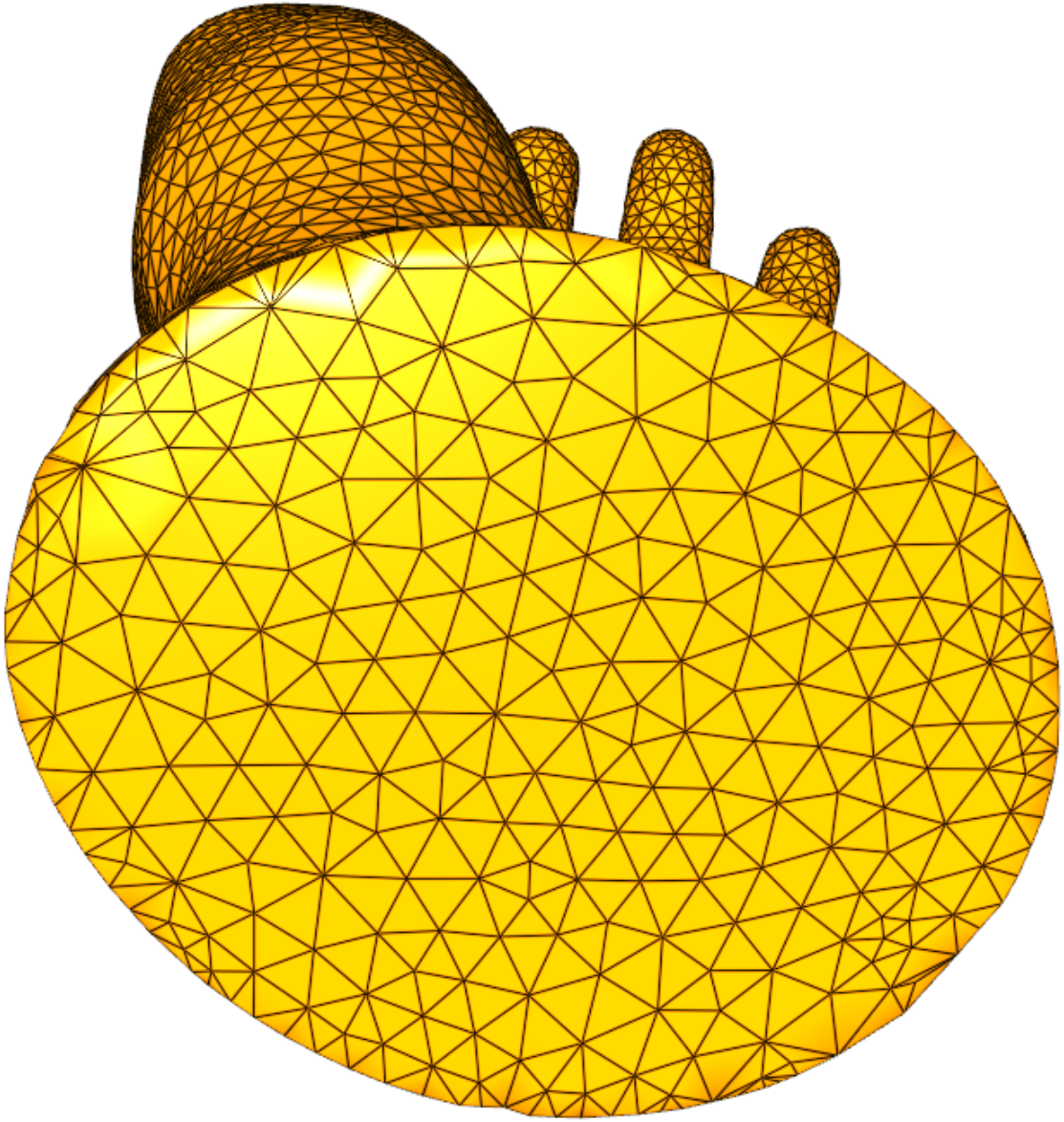


© C.-A. Bohn, FH Wedel, 2011

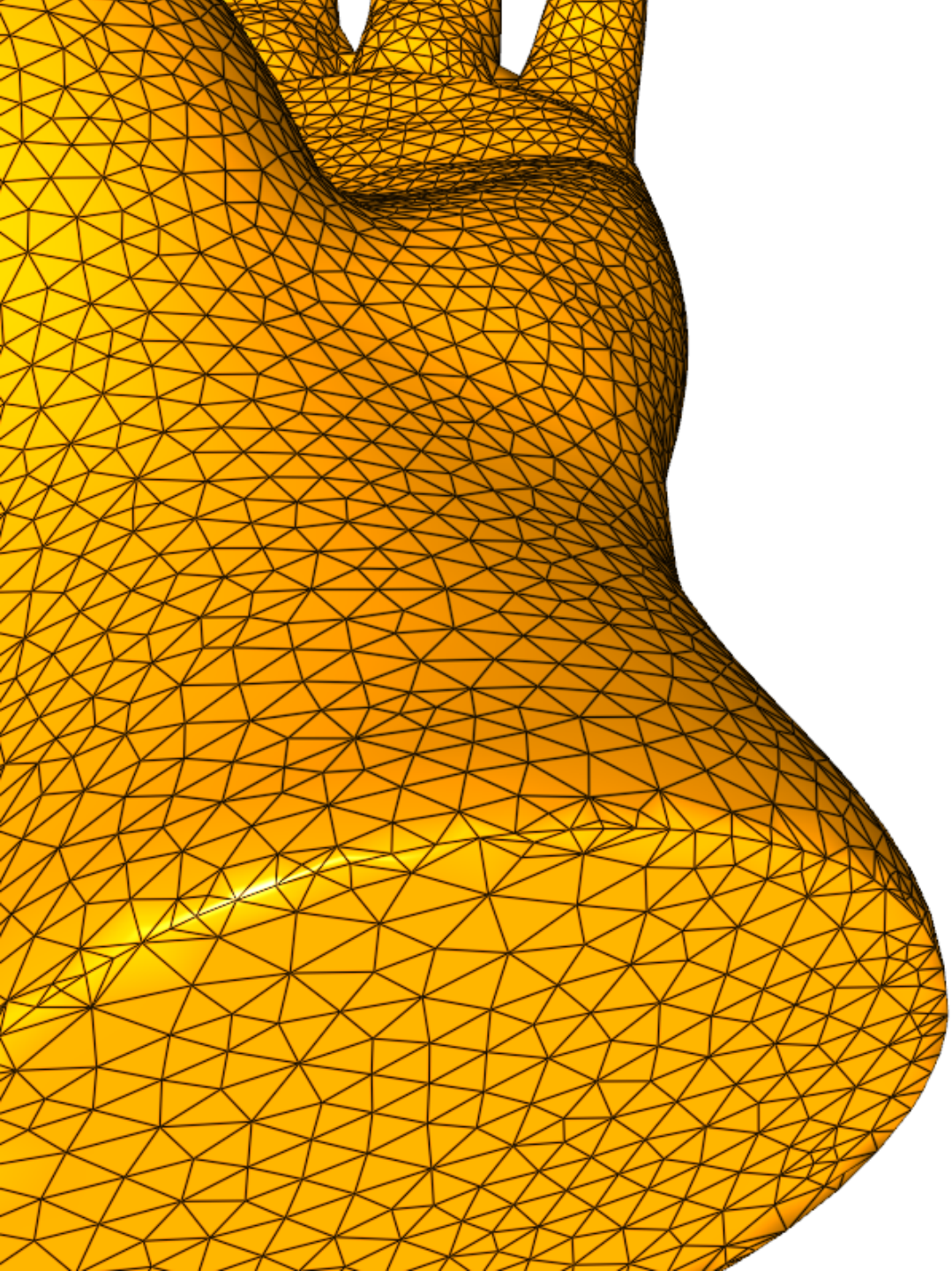


© C.-A. Bohn, FH Wedel, 2011

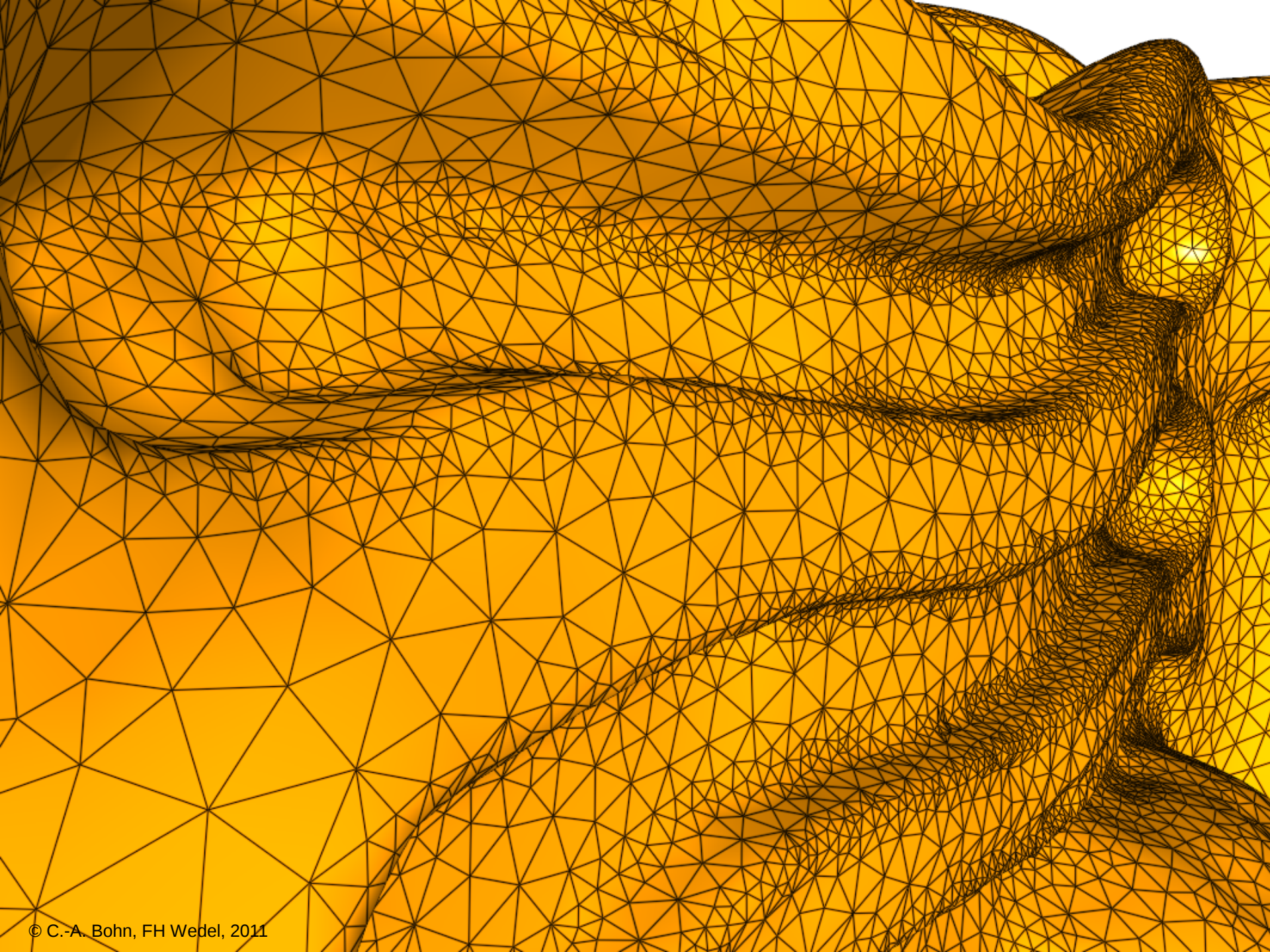
© C.-A. Bohn, FH Wedel, 2011



© C.-A. Bohn, FH Wedel, 2011



© C.-A. Bohn, FH Wedel, 2011



© C.-A. Bohn, FH Wedel, 2011

© C.-A. Bohn, FH Wedel, 2011

© C.-A. Bohn, FH Wedel, 2011

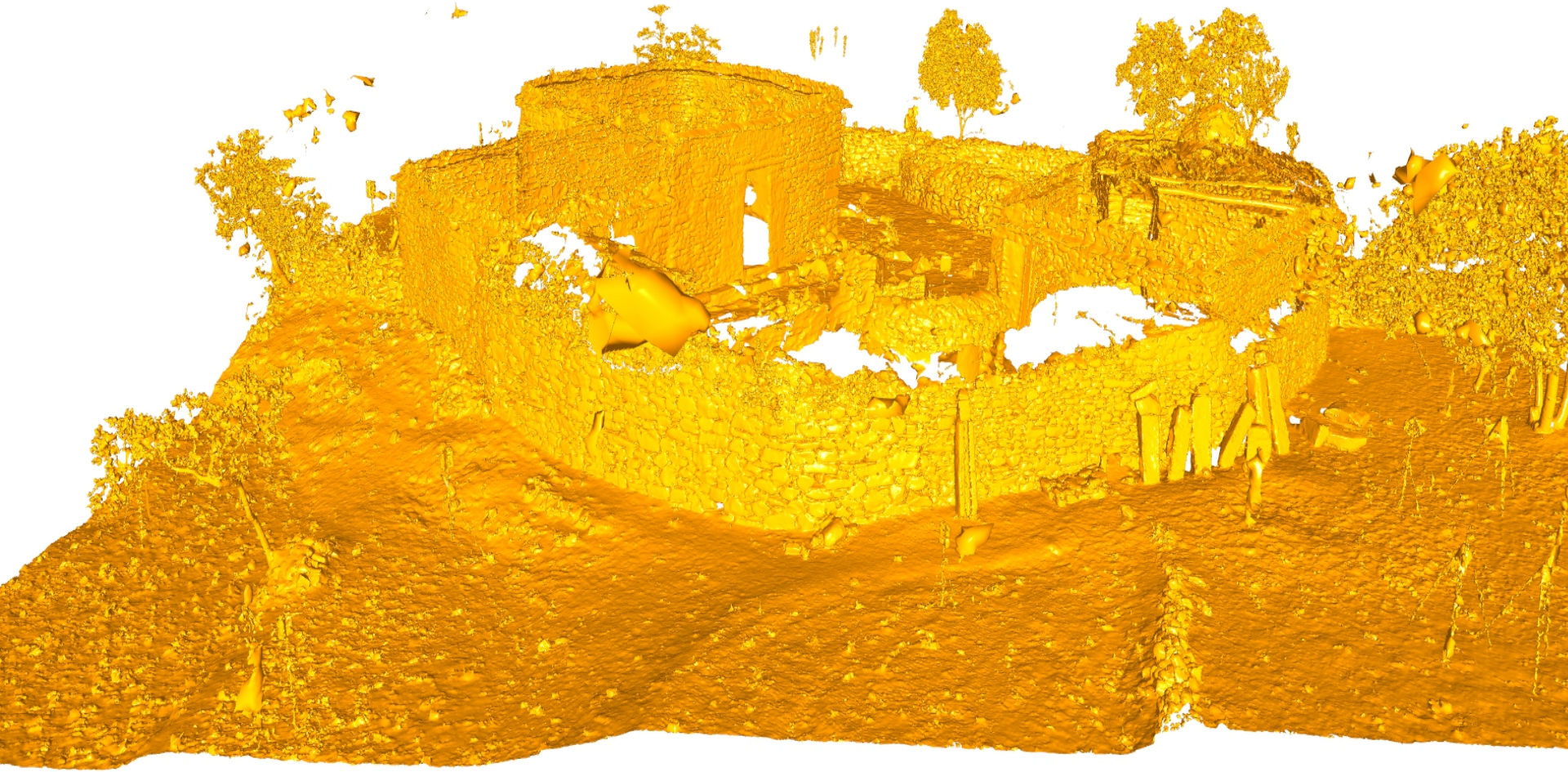
© C.-A. Bohn, FH Wedel, 2011

© C.-A. Bohn, FH Wedel, 2011

© C.-A. Bohn, FH Wedel, 2011

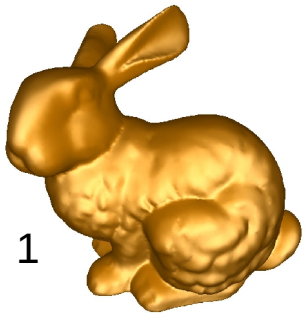
Punktdaten bereitgestellt von FH Düsseldorf

© C.-A. Bohn, FH Wedel, 2011



© C.-A. Bohn, FH Wedel, 2011

Zahlen

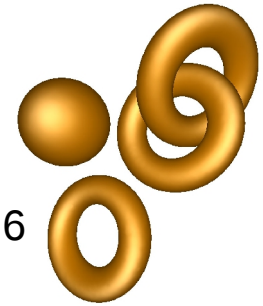

1

2

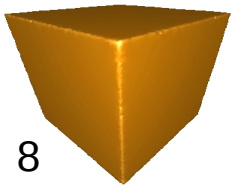
3

4

5


6

7


8

	#Samples	#Cells	Zeit* [mm:ss]	Qualität	Genauigkeit
1	36K	30K	00:39	95,6	4,7e-5
2	438K	100K	02:47	95,5	3,3e-5
3	544K	260K	09:15	93,1	1,7e-5
4	14M	320K	12:17	98,5	1,3e-5
5	5M	500K	21:05	95,9	2,7e-5
6	511K	10K	00:11	99,8	6,6e-5
7	38K	5K	00:06	99	1.5e-5
8	346K	5K	00:06	98,3	0,7e-5

** Intel Core 2 Extreme Quad, 2.53 GHz*

© C.-A. Bohn, FH Wedel, 2011

Zusammenfassung

- Problem: Rekonstruktion ist schwierig, da Spezifikation des Problems und damit ein Algorithmus zu finden schwierig
- Idee: Entwicklung eines Neuronalen Netzes, das es erlaubt, Spezifikation auf beliebigen Ebenen einfließen zu lassen
  => *Supervised* Unsupervised Learning
  => Smart Growing Cells
- Ziel: Eigenschaften der Neuronalen Netze + handgemachte Regeln, die sich dennoch sanft in das Prinzip des ideellen neuronalen Lernens eingliedern.

© C.-A. Bohn, FH Wedel, 2011

Ergebnis

- Löst „jede“ Topologie korrekt auf
- Kann beliebige Mengen von Samples verarbeiten
- Samples können während des Trainings hinzugefügt werden, ohne dass aktuelle Lösung verloren geht
- Jederzeit in der Trainingsphase ein valides Mesh
- Lässt separate Objekte zu
- Produziert direkt eine Triangulierung
- Unempfindlich gegen Rauschen
- Oberflächen-Kurvigkeit wird modelliert
- Faszinierend Robust

© C.-A. Bohn, FH Wedel, 2011

Zukunftige Ideen

- Allgemeine Regeln selbst lernen lassen
  („Supervising Rules for Unsupervised Learning“)
- Remeshing: Level of Detail rückwärts:
  Löschen von Zellen und Neu-Trainieren
- Generalisierung des Konzeptes =>
  Growing Geometry Self-Organizing Network

© C.-A. Bohn, FH Wedel, 2011

Vielen Dank für Ihre Aufmerksamkeit!

Mehr bei...

- Hendrik Annuth: **annuth@fh-wedel.de**
- C.-A. Bohn: **bo@fh-wedel.de**
- Publikationen über SGC:
  http://cg.fh-wedel.de/
  => Research => Publications

© C.-A. Bohn, FH Wedel, 2011